(19) Europäisches Patentamt
European Patent Offic
Office urop´en des br vets

(11) EP 0 855 820 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.07.1998 Bulletin 1998/31

(51) Int Cl.[6]: **H04L 12/56**, H04Q 11/04

(21) Application number: **97480085.6**

(22) Date of filing: **28.11.1997**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **16.12.1996 EP 96480111**

(71) Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION**
Armonk, NY 10504 (US)

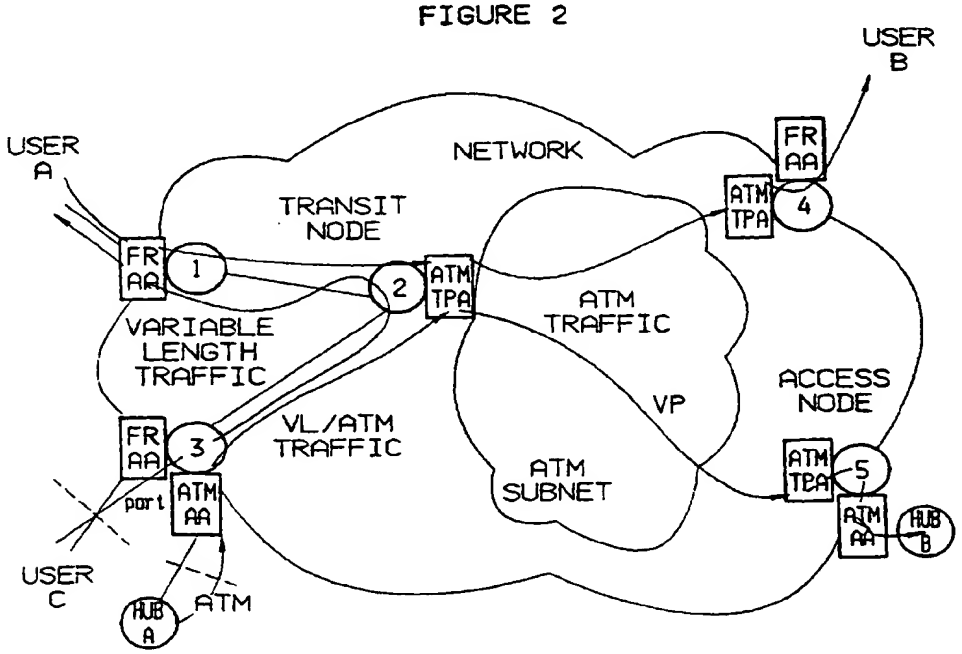
(72) Inventors:
- **Galand, Claude**
  06480 La Colle/Loup (FR)
- **Spagnol, Victor**
  06800 Cagnes sur Mer (FR)
- **Lebizay, Gérald**
  06140 Vence (FR)

(74) Representative: **Therias, Philippe**
Compagnie IBM FRANCE,
Département de Propriété Intellectuelle
06610 La Gaude (FR)

(54) A method and system for optimizing data transmission line bandwidth occupation in a multipriority data traffic environment

(57) This invention deals with data communication networks and more particularly with a method and system for optimizing data link occupation in a multipriority data traffic environment by using data multiplexing techniques over fixed or variable length data packets being asynchronously transmitted. Said packets are split into segments including both a segment number and a packet number. Then the segments are dispatched, on priority basis, over available links or virtual channels based on a so-called global link availability control word indications, which control word is dynamically adjusted according to specific predefined conditions.

FIGURE 2



EP 0 855 820 A2

Description

FIELD OF THE INVENTION.

This invention deals with data communication networks and more particularly with a method and system for optimizing data line/link occupation in a multipriority data traffic environment by using data multiplexing techniques over asynchronously transmitted fixed or variable length data packets .

BACKGROUND OF THE INVENTION

Modern digital networks are made to operate in a multimedia environment for transporting different kinds of digitally encoded data including voice, images, video signals etc..., and enable worldwide coverage while ensuring compliance with a number of requirements specific to each kind of traffics. For instance, while so-called non-real time information can be delivered to the corresponding end-user with minor time constraint restrictions, real-time type of information must be delivered to end-user with predefined limited time delay restrictions.

World-wide coverage is achieved by interconnecting different types of networks including network nodes (i.e access nodes and transit nodes) connected to each other through high speed lines herein also referred to as links. Such a composite network is represented in figure 1. The users get access to the network through ports located in the access nodes. The users' data are processed by an access agent running in the port. The functions of the access agent are two-fold : first interpret the user's protocol, then set the path and route the data through the network.

Different techniques have been developed for organizing the digitally encoded data transport. These include packet switching techniques whereby the-digitized data are arranged into so-called packets. The packets may either be of fixed length, like in the so-called Asynchronous Transfer Mode (ATM), or be of variable length (VL) nature.

A modern international network may then be rather complex, include leased lines and look like the network of figure 2.

In addition to leased lines, this network would support Frame Relay and ATM networks. This network offers the possibility of carrying native Asynchronous Transfer Mode (ATM) traffic as well as Variable Length (VL) traffic, which VL traffic may include both user's traffic and control traffic. A fundamental difference between both VL traffics is that while user's traffic needs be vehiculated along a given path from a source end user to a destination end user without affecting the network, control traffic should be addressed to (a) specific node(s), be decoded therein and control the very network architecture and operation. It should also be noted that whatever be the type of traffic, data are provided to the network at random.

The above helps visioning how complex modern network transmission facilities are. It also helps understanding that the system should be optimized from a cost efficiency standpoint. As per the present invention we shall focus on link operation efficiency, and more particularly optimize link bandwidth occupation.

Let's first recall that links available in the US include so-called T1 operating at 1.544 Mbps and T3 at 44.736 Mbps, while in Europe one may find the E1 at 2.048 Mbps and E3 at 34.368 Mbps.

Now, let's assume a network service provider requiring either an access link to a network or a link between two nodes at medium rate, say between 2 and 10 Mbps. The obvious solution should lead to selecting a T3/E3 link. But this would not fit from a cost/efficiency standpoint bearing in mind the presently practiced tarifs. For example, these tarifs in 1995 are in France (in K$/month) as indicated hereunder :

	50Km	250Km	500Km
E1=	2	10	14
E3=	50	150	170

Accordingly, selecting higher than actually required rates would be prohibitive and one should find a solution for intermediate rates at affordable prices, i.e. optimize bandwidth occupation. A first solution that comes to network designer's mind involves using multiplexing techniques, that is, for instance, cover a 10 Mbps bandwidth requirement with five multiplexed E1 in Europe or seven T1 in the US, or multiplex equivalent virtual channels within a high speed link.

Both hardware and sofware alternatives may be designed. However, hardware solution would suffer the drawback of both high development cost and time to market. Obviously, one should prefer keeping the already available network hardware architectures (e.g.:node architectures) and shoot for software modification.

On the other hand any software implementation of such a multiplexing system should fit with the specific traffic requirements, including quality of service traffic granularity, etc..., in any variable and/or fixed length packet switching network, while requiring minimal development cost for being implemented in already existing network architectures.

OBJECTS OF THE INVENTION

One object of this invention is to provide a method and system for optimizing data communication network bandwidth occupation in a multipriority data traffic environment by simulating a high bandwidth link by multiplexing lower rate links or virtual channels, using software means fitting to already existing network architectures.

Another object of this invention is to provide a method and system for simulating a high bandwidth link by multiplexing lower rate links or virtual channels that is particularly suitable for mixed traffic including both variable length and fixed length types of randomly provided traffics.

Still another object of this invention is to provide a method and system for simulating a high bandwith link by mutiplexing lower rate links or vitual channels enabling random reservation of one of said links or channels specifically assigned a given task.

Still another object of this invention is to provide a method and system for simulating a high bandwidth link by mutiplexing lower rate links or vitual channels, fitting to randomly provided traffics with several predefined Quality of Service (QoS) parameters.

A further oject of this invention is to provide a method and system for simulating a high bandwith link by mutiplexing lower rate links or vitual channels enabling dynamic network bandwidths adaptations through preempt/resume operations.

A still further object of this invention is to provide a method and system for simulating a high bandwith link by mutiplexing lower rate links or vitual channels enabling dynamic network bandwidth adaptation through non disruptive preempt/resume operations.

SUMMARY OF THE INVENTION

The above mentionned objects are achieved by a method for optimizing data transmission link bandwidth occupation in a multipriority data traffic environment, over a data communication network, by simulating a high bandwidth link by multiplexing said traffic over lower rate links or virtual channels, said data communication network including network nodes interconnected by data transmission links, each said network nodes including input and output adapters interconnected to each other through a network switch, said data traffic being randomly provided to the network through fixed and/or variable length data packets, said method including in the node transmission side or output adapter:

- storing said data packets into output queues selected according to a so-called Quality of Service (QoS) based on each said priority levels;

- splitting each said data packets into so-called segments, each segment being provided with a segment header including: a QoS flag defining the corresponding priority level; a packet number reference; a segment number reference; an end of packet flag for identifying the last segment of a processed packet; and validity control bits for header integrity control;

- generating a so-called Link Status Control Word (LSCW) including an at least one bit long flag per link, said flag being used to indicate possible link reservation and thus enable on request link masking;

- generating a so-called Link Availability Control word (LACW) including an at least one bit long flag dynamically settable during operation to indicate whether the corresponding link is currently available or busy;

- performing a logical AND operation between said LSCW and LACW words for generating a so-called global link availability control word;

- monitoring and scanning said output queues on decreasing priority order and multiplexing the segments of said queued packets over said node output links or virtual channels based on said global link availability control word indications.

These and other objects characteristics and advantages will become more apparent from the following detailed description of a preferred embodiment of the invention when considered with reference to the accompanying figures.

BRIEF DESCRIPTION OF THE FIGURES

Figures 1 and 2 are representations of data networks wherein the invention is implementable.
Figure 3 is a schematic representation of a network node output adapter made to include the invention.

Figures 4 and 5 are network parameters to be used for the invention.

Figure 6 is a block diagram of the transmission flow chart for implementing the invention.

Figures 7 through 10 are detailed flow charts for implementing the invention on the transmission side of a network node.

Figure 11 is a block diagram schematically representing the invention on a network node receiving side.

Figure 12 shows a representation of a so-called ring organization to be used for the reception of a given traffic priority.

Figure 13 is a schematic representation of general flow chart for receiving data according to the invention.

Figures 14 through 21 are detailed flow charts for implementing the operations of figure 13.

DESCRIPTION OF A PREFERRED EMBODIMENT OF THE INVENTION

As already mentioned, the invention should be implementable on any network link and therefore it should be controlled from within any network node, be it an access node or an intermediate transit node. In presently available digital networks, each node includes switching means in between receive and transmit adapters. In the preferred embodiment of this invention, the data arranged into packets of either fixed or variable length within the adapters, shall, anyway, be split into fixed length segments (except for the last segment of the packet which might be shorter than said fixed segment length), dispatched through the switching means via switch interface means.

Figure 3 shows a schematic representation of a network node transmit adapter side wherein the packets/segments are provided through switch interface means (300). The segments, if any, may need first being reassembled (301) and then routed toward a queueing means (302) selected according to the specific Quality of Service (QoS) assigned to the processed data.

For illustration purposes, four priority levels based on Quality of Service (QoS) have been defined which include :

- RT1 and RT2 for real time type of traffic with two different relative priorities, RT1 bearing the highest priority level.
- NRT for non real time traffic (e.g. pure data for batch traffic).
- NR for non reserved traffic.

Typically, the highest priority classes (RT1 and RT2) are used to transport voice or video that does not suffer being delayed above a predefined delay. Non-real-time is used to transport interactive data. And non-reserved traffic, used for file transfer for instance is assigned the lowest priority level. In addition, some control data may require being forwarded to a general purpose processor (GPP303) controlling some of the network operations. These control data may therefore just leave the transmission path at the considered node level.

The above mentioned priority criteria, while they complicate the invention mechanism are of a high interest for the invention. In other words they shall introduce important constraints in the final method to be developed. For instance the queues shall be served in the scheduler with so-called preempt/resume function facilities.

Accordingly, the packets enqueued therein are split into segments, each segment being assigned a segment number between zero and N, N being equal to the maximum expected packet length divided by the predefined segment length according to the required bandwidth. Now, a scheduler (304) cooperating with a transmission group segmenting device (305) shall sequentially scan the considered packet queued, segment these packets and assign each of these segments an available output line/virtual channel L1,L2,...Ln. Preferaby all multiplexed links should be configured at the same transmission rate. To define the channel availability a n-bits long so-called Line Availability Control Word (LACW) is defined, with each bit position being dynamically set either to one to indicate that the corresponding link (channel) is available, or to zero for unavailable link, i.e. for a channel being currently active, i.e. transmitting data. Also, any of the n link/channels must be momentarily reservable for a specific assignment, be candidate for a preemption operation or be physically dropped, etc.... Therefore the Link Availability Control Word (LACW) shall be masked by a so-called n-bits long Line Status Control Word (LSCW). Each LSCW bit position is either set to one for enabling conditions or to zero for disabling. In the preferred embodiment of this invention, the LSCW shall be handled and already existing network facility assigned to line resources Management, while the LACW shall be handled by the transmission program to be described herein.

In operation, LACW and LSCW may handle groups of trunks (physical and/or logical) instead of a same trunk. For instance, a first part of a given LACW/LSCW might belong to a first group, while the second part would belong to a second group . In this case, each group defines an aggregate trunk. At the system transmit side, the group is selected thanks to a routing table for a given packet connection identification. Accordingly, a so-called group numb r (gn) parameter should be added to the segment header.

Finally, a global link availability for the multiplexing operations is defined through logical ANDing of both LACW with LSCW control words.

Also, in practice, each of the n links might be taken at random based on availability criteria over the total line or

trunk bandwidth. Accordingly, the LACW might be longer than n, and the required n links/channels shall be selected at random over the total available channels.

In operations, the processed data packet is split into segments and each, segment shall include a 4 bytes long header and 60 bytes (or less in case of the last segment of a processed packet) be reserved for the segment payload. In the preferred embodiment, the segment header includes :

- QoS : 2 bits coding RT1, RT2, NRT and NR, i.e; 00 is for RT1, 01 for RT2, 10 for NRT, and 11 for NR.
- Packet number : 7 bits for coding the packet number.
- Segment Number 6 bits for coding the segment number
- L : 1 bit set to one to indicate whether the considered segment is the last segment of the considered packet.
- Time stamp : 15 bits for time stamping and transmit time and header integrity check.

By comparing the "L" bit value and corresponding segment number, the system shall be able to check whether the packet was fully received or not.

Let's now focus on the transmit operations as implemented in the preferred embodiment of this invention.

To that end, some facilities shall be defined. These include first a so-called Transmit (XMIT) Status Byte organized as represented in figure 4. Let's assume the XMIT Status Byte is scanned from bit position zero to bit position seven. Each bit position is used either to define a so-called Transmit In Progress (XIP) flag or a so-called Transmit Queue (XQ) flag. Each Quality of Service (QoS) is assigned two consecutive bit positions, one for XIP and one for XQ. The XIP flag bit being OFF indicates that no packet transmission is in progress; it is set ON to indicate that a packet transmission has been started, i.e. at least its first segment is transmitted. The XQ bit being OFF indicates that the related QoS XMIT queue is empty, while it shall be ON when at least one packet is stored in corresponding transmit queue.

Said queues have been coded as follows :

- RT1	Real Time 1	(QoS "00") for highest priority
- RT2	Real Time 2	(QoS "01")
- NRT	Non Real Time	(QoS "10")
- NR	Non Reserved	(QoS "11") for lowest priority

Also, a Packet Transmit Control Block (PXCB) including six bytes per Qos is used to keep record of the numbering of the corresponding packets and segments as well as pointers to packet list. Said PXCB is shown in figure 5. It should be noted also that the packet list is a list of sixty segment packet list. Said PXCB is shown in figure 5. It should be noted also that the packet list is a list of sixty segment pointers, each pointing to a buffer where the data of the corresponding packet segment have been received from the node switch and stored. Each buffer has been made sixty four bytes long, sixty bytes being used for segment data and four bytes for segment header, said header bytes position being initially empty. The XMIT Packet List sixty through sixty three are used for storing packet information such as: packet routing information, packet length, packet assignment such as network control, etc..

Let's now proceed with describing the flowcharts detailing the transmission operations as implemented in the preferred embodiment of this invention. Given these flowcharts, a person skilled in the programming art shall be able to write the programs driving the transmission process without any inventive effort being required from his part.

Represented in figure 6 is the general flow diagram for the transmit operations. It includes both high priority tasks for End Of Segment (EOS) interrupt program (601), and low priority tasks (602) for the main steady state program. At End Of Segment interrupt, thanks to the link index (Ix), the related Qx value allows to reset the related link active flag and release the corresponding segment buffer, before returning to the main steady state program. This program starts with (603) getting a packet from the Switch and enqueuing it in the corresponding QoS transmit queue. A test is then performed (604) to detect whether a link is available. If not, the process loops back to start at (603), otherwise, it should proceed (605) to dequeue a segment from the highest QoS transmit queue and calculate the segment header (i.e. the already mentioned four bytes) for feeding the header in the buffer pointed at by the segment pointer of the transmit packet list. Finally, link selection operations are performed for selecting a link among the available links and setting the corresponding link active flag; And then, start segment transmission (606) prior to looping back to start (603).

The flow chart for the End Of Segment interrupt program is represented in figure 7. First (701) a link index (Ix) is provided by hardware means for addressing the Link Xmit Control Block (LCB). Then (702) the process resets the Link Active flag accordingly, by setting the LACW bit(x) OFF; and then release the transmitted segment buffer, prior to going back to the main steady state program.

Said main steady state program operations start (see figure 8) with checking whether a packet has been received from the node switch (801). If yes, then said packet is enqueued (after getting connection identity from the packet

header and fetching group number gn from a routing table) into one of the queues based on the packet QoS parameter. This operation might, naturally be combined with any existing mechanism for regulating the queues levels. Once a packet is enqueued, the XQ flag bit is set ON in the transmit status byte to indicate the presence of said packet in the queue (802). A test is then performed (803) on a so-called global link availability control word obtained by ANDing the Link Status Control Word with the Link Available Control Word, to check whether a link is available for transmission. Said link availability test (803) is also performed in case of negative answer to test (801). Otherwise the first available link is selected and its index is recorded (804).

The process goes then to XP1 (see figure 9) representing the transmit status byte which is scanned in decreasing priority order, i.e. from QoS=00 to QoS=11 and more precisely from bit range zero to seven in the XMIT Status Byte. For each priority two sets of operations might be performed. Each set is represented by a block (see 901 and 902). Block 901 indicates a packet of related QoS is transmitting. It includes preparing Qx index for procesing the related PXCB (continuing transmission) and then branching to XP2B. block 902 indicates that at least one packet is ready for being transmitted from related QoS. A Qx index is prepared for processing related PXCB (starting transmission of a packet). Additionally, if the queue, after extracting, is empty, XQ flag is set to zero. Naturally if scanning the transmit status byte in decreasing priority order fails to stop on any of said priorities, the process exits, i.e. goes back to (801), otherwise it branches either to (XP2A) for a new packet, or to (XP2B) for a new segment being transmitted (see figure 10).

As represented in figure 10, for a new packet, the related XIP bit is set ON, the packet sequence number as indicated by the XPKTN counter is incremented for the considered priority queue (1001). Then the segment sequence number counter for said considered priority is initialized (1002). If the packet transmission already started, (XP2B entry) the process would only need incrementing the segment sequence number counter (1003). Then, in both cases, that is either after (1002) or after (1003) the process goes to (1004) for dequeueing a segment pointer from the selected packet list XPKT and starts (1005) constructing the four bytes long segment header accordingly. The quality of service field of said header is filled with the considered QoS; the packet number and segment number fields are filled with the content of the packet number counter and segment number counter fields, respectively. And the bit L field defining last segment of packet is either set to zero or to one accordingly, L=1 identifying a last segment. Finally the header integrity control byte is calculated by XORing all other header bytes as already mentioned.

Transmit operations proceed then (1006) with starting transmission position pointed at. The link active flag is set ON in the Ix position of the Link Active Control Word.

A test (1007) is then performed to detect whether the current segment is the last segment of the packet being transmitted. If not, then the process loops back to start (see figure 8).

Otherwise, a new packet shall be made ready for transmission in the considered Quality of Service and, to that end, the XIP bit is set OFF in the transmit status byte accordingly.

Let's now describe receive operations as implemented in the preferred embodiment of this invention and summarized in the block diagram of figure 11. First, the receive process should properly resequence and reassemble (1101) the segments into their original packet form. This part is the most complex of the receive process. Then proper routing (1102) toward the already mentioned General Purpose Processor (1103), for control traffic data for instance, or toward the Switch Interface 1104. Such a process shall match perfectly with the existing node architecture as disclosed in the copending European Application, filed on July 20, 1994 with title "Multipurpose Packet Switching Node for a Data Communication Network", Publication Number 000079065.

The basic principle of the receiving operations is represented in figure 12 for one Quality of Service (QoS). In other words, the complete system shall be made to include as many of the represented elements as there are defined QoS, e.g. four for the best mode of implementation, i.e.: RT1, RT2, NRT and NR. Each QoS is thus provided with a ring (1201) split into 128 ring elements for packet status (PKT-STAT) which ring elements are numbered sequentially zero to 127.

Each ring element is associated a buffering zone named Packet Structure and packet information mapped into the Packet Structure. Each ring element contains also so-called Status Flags including :

- RIP : Reassembly In Progress flag which is set ON when receiving a first segment (Nota Bene : not necessarily the segment numbered zero, since segments could be received at random) of a new packet. Said RIP flag is set OFF when flushing the received packet for any reason including error or transfer of complete packet from the Packet Structure into a corresponding Switch Output Queue (SOQ).

- FLUSH : Flush flag is set ON when for example a time-out is detected (time-out parameters shall be defined to avoid jamming the system with uncomplete packets due to any kind of transmission failure). Flush flag is set OFF when the first segment of a new packet for the same ring element arrives as shall be indicated by paramet rs including so-called RCF variable.

- PKT-COMPLETE : This flag is set ON when all segments of a same packet have been received as indicated by counters including so-called SEGCTR and EXPCTR.

- EOP : End Of Packet flag is set ON when last segment of a packet (L flag ON) has been received.

A built in connection is designed for relating any given Ring Element to a Packet Structure. The Packet Structure zone has been made to include sixty four, 4-bytes long, slots : sixty slots (i.e numbered zero to fifty nine) being reserved for the segments and four slots (sixty to sixty-three) for packet information. Said packet information include :

- the segment counter (SEGCTR) counting the number of segments received for a same packet.

- the expected number of segments counter (EXPCTR) in a packet. EXPCTR is set ON when receiving the segment marked "L" (last) in the 4-bytes long segment header. The segment sequence number of this segment indicates how many segments should be received for the considered packet. When SEGCTR=EXPCTR, the packet is complete.

- A Ring Cycle value at packet flush time (RCF) is initialized at so-called flush flag setting ON time, and used for distinguishing a flushing packet from the new one which should occupy same ring element (same packet number modulo 128).

In fact, the receive Packet Structure is addressed by the packet number index (px) pointing to related packet ring element. Said index is formed with the seven bits long packet sequence number value extracted from the four bytes long segment header. The oldest packet, as indicated by the lowest packet number in Receive In Progress (RIP) state is referred to as base packet and is identified by its packet index (bpx). The receive Packet Structure is made to store segment pointers only, which pointers shall point to a buffering zone within a RAM of the node adapter. Each segment pointer is addressed by a slot number or segment index (Sx) formed with the six bits long segment number value extracted from the four bytes long segment header. When empty, a slot contains zeros. The RAM buffering zone shall enable storing at least 60*60 bytes = 3600 bytes for the packet data bytes.

Once a packet is complete, the above described PKT_STRUCT should be cleared for receiving next packet while the system might not be ready for passing the completed packet to the network node switch. Accordingly, the buffering means (PKT_LIST) is used wherein the corresponding packet information are transferred. These include the general packet information such as destination, length, etc... and the first to last segment pointers pointing to the RAM zone storing the segment payloads for next transmit queueing.

The general flow diagram for receive operation is schematically represented in figure 13. It includes both low priority operations for the main steady state program (1301) and high priority tasks (1302) for processing end of segment interrupts. At end of segment (EOS) interrupt, a received segment is properly enqueued and time stamped (1303). Properly enqueueing means identifying the QoS involved and addressing the receiving system including ring and associated elements accordingly. Then the buffer pointer is enqueued in the Link Input Queue (LIQ) (1304) together with a time stamp, i.e. time of day (TOD) provided by the system timer at each end of segment received. Then a new buffer is prepared for new segment (1305) prior to the process branching back to the main steady state program (1301).

Basically, said steady state program shall enable performing three basic sets of operations, namely :

- getting a segment pointer and its time stamp from the line input queue (1305);
- reassembling the packet and controlling any possible time-out (1306); and,
- routing and transmitting the complete packets, e.g. towards the node switch (1307).

The high priority tasks (1302) are implemented thanks to the flowchart of figure 14. The process starts with using the actual time indicated by the system timer as Last Global Time Stamp (LGTS) (1401). The LGTS is thus refreshed with the internal timer value every time a segment is received, i.e. when branching to the EOS interrupt program. Then (1402) the received segment is properly enqueued into the Line Input Queue after being provided with a segment pointer and a receive time stamp. Then a new buffer is prepared for next segment of the considered packet (1403) and the process branches to the main steady state program. Naturally any errored segment should have been discarded.

The main steady state program is implemented in the flow charts of figures 15 through 21 detailing th process of operations 1305 through 1307.

As represented in figure 15, after entry point PR_ENTRY, the process starts with checking the line input queu for highest priority, and then in decreasing priority order, to detect whether said checked queue is empty (1501). Should it be empty, then the flow chart branches to PR4 to be described later on. If at least one segment is present in the

queue, said segment is dequeued and its segment pointer is stored as Last Received Segment Pointer pointing to the 64 bytes buffer where the segment data have been stored. Also, the dequeued segment is set and stored as Last segment Time Stamp (LTS) (1502).

Next step (1503) involves setting internal program variables from the 4-bytes long segment header, after checking said header validity thanks to the fourth header byte, and discarding wrong LRC checked segments. For valid segments processed, said variables involve a packet index (px), a segment index (sx), a sequence number (LSN) and a ring index (rx). The packet index shall be used for pointing to related packet ring element and Packet Structure and Packet Status. The packet index is formed with the seven bits long segment header. The segment index (sx), used for pointing to the related Packet Structure slot number, is formed with the six bits long segment sequence number value extracted from the four bytes long segment header. The ring index (rx) is formed with the two bits long quality of service parameter extracted from the four bytes long segment header. Finally, a Last Segment Sequence Number (LSN) is detected through concatenation of the six bits of the segment sequence number to the seven bits of the packet sequence number.

The ring index (rx) made to select among rings RT1, RT2, NRT and NR is used to memorize the value of QoS of last received segment and shall be used, later on for performing so called packet time-out determination (see figure 20). While the LSN and LTS are used as Last Received segment Sequence Number and Last received segment Time Stamp, respectively (see 1504).

Then, the highest received sequence number of ring is updated. To that end the Last received Sequence Number is compared to the Highest Received Segment Sequence Number (1505). Should LSN be lower than HRSN(rx), then no updating is necessary. Otherwise the HRSN(rx) and its Time Stamp are updated accordingly (1506) and the process branches to routine PR1 of figure 16.

As represented in figure 16, the routine starts with checking whether the flush flag for the considered ring index and packet index is ON. If yes, then the system is in flushing packet status (to be detailed later on in this description) and all segment belonging to the considered packet shall be discarded. To that end, the process checks whether the considered packet is a new packet (1602) (i.e. same packet number +128), if not, then the segment buffer is released (1603) due to continuing flushing condition for considered ring slot, and the process branches to PR4. Otherwise, the flush status is considered old and a new packet needs be processed, the process branches the same routine as the one following an OFF flush flag detection. Next test (1604) tests whether the ring is empty, in other words no ring slot is occupied in the ring. If yes, as indicated by the counter counting the number of packets in reassembly state as marked by a RIP flag ON in the considered ring, then the process branches directly to PR2. Otherwise another test (1605) is performed to check whether the current packet index, modulo 128, is lower than the Ring Base Packet.

In other words, it should be noticed that at any time there is an oldest packet in reassembly status in each ring, which packet is called Base-Packet of the ring. As long as this packet is not complete, all subsequent packets, even if complete, cannot be sent to the node switch for preventing from packet desequencing. The base packet index needs, naturally, be updated, then if px<RBPT(rx), px replaces the Ring Base packet Pointer (RBPT(rx)) (see 1605). Then the routine branches to PR2, as it also does for a negative (1605) test. The base packet has been updated and the ring origin is known.

The PR2 routine (see figure 17) starts with checking whether current received segment starts a new packet or not, by testing the RIP(rx,px) flag (see 1701). If a new packet is starting, then the designated Packet Structure should be prepared (1702) by emptying the segment slots and setting the Packet Status flags to zero. The system may then update the counters accordingly (1703) and therefore after setting the RIP flag ON, the count of packets in reassembly state in the considered ring is incremented and the segment count for the considered packet is reset to zero. Now, should test (1701) indicate a flag ON, or once the operations (1703) are performed, a test (1704) is performed to check whether the Last received segment Sequence Number, obtained by concatenating the six-bits long segment sequence number to the seven-bits long packet sequence number is equal to the Expected segment Sequence number, i.e. the lowest segment number not received yet, that is still being waited for, for a base packet. If the test (1704) is positive, then this expected segment was just received. Then RESN is refreshed (1705). Accordingly, RESN(rx) is set to the value of next empty slot in the Packet Structure for (rx,px). Then the routine branches to PR3, as it also does when the result of test (1704) is negative, which indicates that the expected segment (lower segment number) was not received yet.

The PR3 routine is represented in figure 18. This routine starts (1801) with storing received segment pointer in the Packet Structure, as already defined, then update the count of number of segments received for considered packet, and update last received segment pointer. Then the End Of Packet (i.e."L") flag is tested (1802). If this flag is set identifying a last segment, the system knows then the number of segments that should have been received for the considered packet. The expected segment counter is loaded accordingly. The End Of Packet flag bit is s t ON (1803). If the EOP flag is ON (1804) and the segment counter count is equal to the Expected segment counter count (1805), then the packet is complete. Accordingly, the flag identifying a complete packet is set ON (1806) and the process branches to PR4. Otherwise, should tests (1804) or (1805) be negative, the routine branches also to PR4 for clearing the ring and associated structure for complete packets or timed-out packets, that is uncomplete packets which have

been waiting too long (i.e. over a predefined time threshold) in the ring.

Represented in figure 19 is the PR4 routine. This routine shall address the four priority rings, by setting rx to zero (1901) and incrementing it progressively (1902) up to the highest ring, e.g. rx=3. If the ring is empty, as indicated by test (1903), then the routine issues to PR5 as soon as rx=3. If the ring is not empty, px is set to the ring base pointer (1904). The flag indicating a complete base packet is tested (1905), since as long as the base packet is uncomplete, and unless time is out, the packets younger than the base packet are not transferred to the node switch. Then, assuming the flag indicating a complete base packet is ON, the base packet structure is transferred into the Packet-List, and the Packet Structure and Status are cleared (1906). Then the packet is enqueued (1907) into the Switch Output Queue.

But should the flag, as tested in (1905) be OFF then a test (1908) on uncomplete packet is performed to detect whether a time-out condition occured. (This parameter shall be described with reference to Figure 20). If not, the routine branches to next ring as long as rx is not equal to three (1910). Otherwise, in case of time-out, then (see 1911) the segment pointers are released accordingly and the ring slot is emptied; the Packet-Status is set to zero and the flush flag is set ON accordingly.

Then, the count of number of packets in reassembling state in the ring, as marked by a RIP flag ON, is decremented (1912). Same operation is also performed after (1907). Said count, once decremented, is tested versus emptiness. Should the ring be empty, the routine branches to (1910), otherwise a reinitialization of base packet (i.e. actual base packet) in reassembling state in the ring (1914). The ring base packet pointer is set and the index of first empty slot (lowest segment number of segment not received yet) belonging to the base packet, is detected and its index stored.

As already mentioned system jamming has been made avoidable by defining time threshold(s) which should not be exceeded during packet receiving and assembling. A corresponding so-called time-out control mechanism is designed accordingly, as represented in the flowchart of figure 20.

To help undestanding the time-out mechanism operation, let's first define the rules selected in the preferred embodiment of this invention, for determining that a packet has been waiting in the ring for too long and should be discarded. Accordingly, the following rules shall apply :

1)- A segment "n" should arrive at destination before :

Time Stamp of segment "n+1"+RSTO wherein RSTO, i.e. Received Segment Time Out is based on the addition of :

- theoritical propagation delay over one trunk (T1/E1 for instance) of a 64 bytes long segment of packet; plus,
- a correction value, i.e. a tolerance, due to the different transmission rates on each trunk, or clock variations.

This means that, when a segment "n+1" has arrived, the segment "n" should not be too late. For instance, let's assume a 64-bytes long segment "n" is sent over link 1, then a 5 bytes long segment "n+1" is sent over link 2. Segment "n+1" shall arrive before segment "n" but the maximum delay for receiving segment "n" should be equal to the Transmission delay for a 64 bytes long segment + correction value DELTA (predefined value).

2) Due to preemption rules applying on the transmit side for QoS priority purposes, a packet may stay in Receive In Progress, and not complete, state for a time higher than RSTO when higher QOS segments have been interleaved in the transmitted traffic. Accordingly, in case of reception interruption for a predefined, say 2 RSTO for instance, packets in RIP status and not complete should be flushed.

Let's now consider the Time-Out Control flowchart of figure 20 detailing the operations performed at step 1905 in figure 19. First (see 2001) if current time as read in the system timer at end of segment reception if greater than Last Global Time Stamp plus a predefined time threshold, herein selected for being equal to two times the so-called Receive Segment Time Out value, then no reception occured for a while and time-out is verified. The routine exits. Otherwise a new test is performed (see 2002) testing for inter-ring time-out. Normally, when reception switches to a lower priority, then the higher priority reception should be over. Then (2002) is made to check whether the reception switched to a lower priority or not while still expecting segments. If it did switch, then the time-out checking should be performed (2003) on the Last Received segment Time Stamp on ring rx, with the threshold set now to RSTO. If this test is positive, indicating that a low priority segment was received before getting rid of higher priority, then time-out occured. Otherwise, the routine branches to a new test (2004), as it also does for a negative (2002) test. The test (2004) is made for intra-ring time-out control. It checks whether the lowest segment of a given sequence number not yet received in a base packet in ring rx is older than the highest received segment sequence number, said sequence numbers being defin d by the concatenation of the six bits long segment number to the seven bits long corresponding packet number. If not, then no time out occured. Otherwise a time-out check is to be performed (2005). In other words, a positive test (2004) indicates that a time-out rule should be applied (i.e. 2005). If the Time Of Day (TOD) is higher than the predefined time-out parameter (RSTO) added to the receive time stamp (HRTS) of the Highest Received segment Sequenc Number

as defined by the concatenation of highest received segment number appended to the corresponding packet number, then the expected segment has not been received in time, i.e time is out. Otherwise as also when test (2004) is negative, no time-out occured and the process may go on.

The PR5 routine, as represented in figure 21, is meant to terminate the receive process as performed in the network node receive adapter and forward the received packet toward the node switch. To that end, first the content of the Switch Output Queue (SOQ), wherein the complete packets extracted from the ring have been enqueued, is tested (2101). If said queue is empty, then the process branches directly to PR_ENTRY (see figure 15). Otherwise the packet list is dequeued (2102) and the packet is properly routed (see figure 11) (2103) and the packet is sent to the node switch (2104) before branching back to main steady state receive process entry.

Given the above detailed flowcharts of both transmit and receive operations as applying to the present invention, a person skilled in the programming art shall have no difficulty in implementing the invention without any additional inventive effort being required Also, as already mentioned, a valuable advantage of this invention derives from its easyness of implementation in already available network node, with little additional software means and almost no additional hardware being required. Therefore, this invention is particularly valuable from cost efficiency standpoint, and of high business interest in the present multimedia environment.

Claims

1. A method for optimizing data transmission link bandwidth occupation in a multipriority data traffic environment, over a data communication network, by simulating a high bandwidth link by multiplexing said traffic over lower rate links or virtual channels, said data communication network including network nodes interconnected by data transmission links, each said network nodes including input and output adapters interconnected to each other through a network switch, said data traffic being randomly provided to the network through fixed and/or variable length data packets, said method including in the node transmission side or output adapter:

   - storing said data packets into output queues selected according to a so-called Quality of Service (QoS) based on each said priority levels;
   - splitting each said data packets into so-called segments, each segment being provided with a segment header including: a QoS flag defining the corresponding priority level; a packet number reference; a segment number reference; an end of packet flag for identifying the last segment of a processed packet; and validity control bits for header integrity control;
   - generating a so-called Link Status Control Word (LSCW) including an at least one bit long flag per link, said flag being used to indicate possible link reservation and thus enable on request link masking;
   - generating a so-called Link Availability Control word (LACW) including an at least one bit long flag dynamically settable during operation to indicate whether the coresonding link is currently available or busy;
   - performing a logical AND operation between said LSCW and LACW words for generating a so-called global link availability control word;
   - monitoring and scanning said output queues on decreasing priority order and multiplexing the segments of said queued packets over said node output links or virtual channels based on said global link availability control word indications.

2. A method for optimizing data transmission link bandwidth occupation in a multipriority data traffic environment by simulating a high bandwidth link by multiplexing said traffic over lower rate links or virtual channels according to claim 1, wherein said monitoring and scanning of the output queues on decreasing priority order for multiplexing the data segments of enqueued packets includes:

   - defining a so-called Transmit Status word split into consecutive pairs of bits, each bit of said pairs being used as a so-called Transmit In progress (XIP) flag or a so-called Transmit Queue (XQ) flag, each pair of flags b ing assigned to a QoS in decreasing priority ord r;
   - setting the XIP flag ON to indicate that a packet transmission of the corresponding priority has been started, or setting said flag OFF to indicate that no corresponding packet transmission is in progress;

- setting the XQ flag OFF to indicate that the related QoS transmit queue is empty, or setting said flag ON when at least one packet is stored in the corresponding priority queue;

- scanning the Transmit Status word in decreasing priority order to enable controlling the transmission by assigning node output links to the non-empty queue of highest priority.

3. A method for optimizing data transmission link bandwidth occupation in a multipriority data traffic environment by simulating a high bandwidth link by multiplexing said traffic over lower rate links or virtual channels according to claim 2, said method further including:

- generating a so-called group number (gn);

- using said gn for splitting said LSCW/LACW control words accordingly,

whereby several sets of aggregate trunks might be defined accordingly.

4. A system for optimizing data transmission link bandwidth occupation in a multipriority data traffic environment over a data communication network, by simulating a high bandwidth link by multiplexing said traffic, randomly provided to the network nodes in the form of fixed or variable length packets, over lower rate network links or virtual channels, each said network nodes including input and output adapters interconnected to each other through a network switching means, said system including in the network output adapter:

- a switch interface (301) interfacing the node switch to the output adapter means;

- routing means (301) for routing switch provided data packets either towards processor means (303) for processing network control data or toward a priority organized queueing means (302) selected according to a predefined Qality of Service (QoS)defining a priority assigned to the processed data;

- scheduler means (304) cooperating with segmenting means (305) and including:

- means for splitting each said data packets into so-called segments, each segment being provided with a segment header including: a QoS flag defining the corresponding priority level; a packet number reference; a segment number reference; an end of packet flag for identifying the last segment of a processed packet; and validity control bits for header integrity control;

- means for generating a so-called Link Status Control Word (LSCW) including an at least one bit long flag per link, said flag being used to indicate possible link reservation and thus enable on request link masking;

- means for generating a so-called Link Availability Control word (LACW) including an at least one bit long flag dynamically settable during operation to indicate whether the corresonding link is currently available or busy;

- means for performing a logical AND operation between said LSCW and LACW words for generating a so-called global link availability control word;

- means for monitoring and scanning said output queues on decreasing priority order and multiplexing the segments of said queued packets over said node output links or virtual channels based on said global link availability control word indications.

5. A method, according to claim 1, for optimizing data transmission link bandwidth occupation in a multipriority data traffic environment over a data communication network, by simulating a high bandwidth link by multiplexing said traffic over lower rate links or virtual channels, said data communication network including network nodes interconnected to each other by data transmission links, each said network nodes including input and output adapters interconnected to each other through a network switch, said data traffic being randomly provided to the network through fixed and/or variable length data packets, said method including in the node receiving side or r ceive adapter:

- monitoring and scanning the node input links/channels for collecting the received segments, and time stamping these;

- resequencing these segments and properly reassembling these (1101) into the original transmitted packets, based on information recorded into said segment headers and including the corresponding packet number and segment number;

- routing the reassembled packets (1102) eather toward a node processor (1103) for network control packets, or toward the node switch for further transmission (1104).

6. A method according to claim 5 wherein said segment resequencing and packet reassembling include:

- defining a so-called ring structure per processed priority level, said ring being split into consecutive ring elements;

- sequentially assigning the ring elements to the received packets and storing in each said ring element a so-called Status flag;

- associating to each ring element a so-called Packet Structure for storing therein segment information according to the corresponding segment numbering, and Packet Information relating to the considered received packet.

7. A method according to claim 6 wherein said Status flags include:

- a reassembling In Progress flag set ON when receiving a first segment of a new packet, and set OFF when clearing the ring element;

- a so-called Flush flag set ON when a predefined time-out threshold is reached while packet reassembling is not achieved yet, and set OFF when receiving the first segment of a new packet for same ring element;

- a Packet Complete flag set ON when all segments of a same packet have been received; and,

- an End Of Packet flag set ON when the last segment of a packet, as indicated by the segment header, has been received.

8. A method according to claim 6 or 7 wherein said so-called Packet Information relating to a considered received packet include:

- a segment count, counting the number of segments received for a same packet; and,

- an expected number of segments count in a packet as indicated by the segment sequence number in the header of the segment identified as last segment of a packet;

whereby a packet being received may be considered complete when the segment count equals the expected number of segments, thus enabling the routing and transfer of said packet toward its final destination.

9. A method according to claim 7 wherein said Packet Structure is made to store segment pointers, each pointing to an assigned buffering location where the segment data are actually being stored.

FIGURE 1

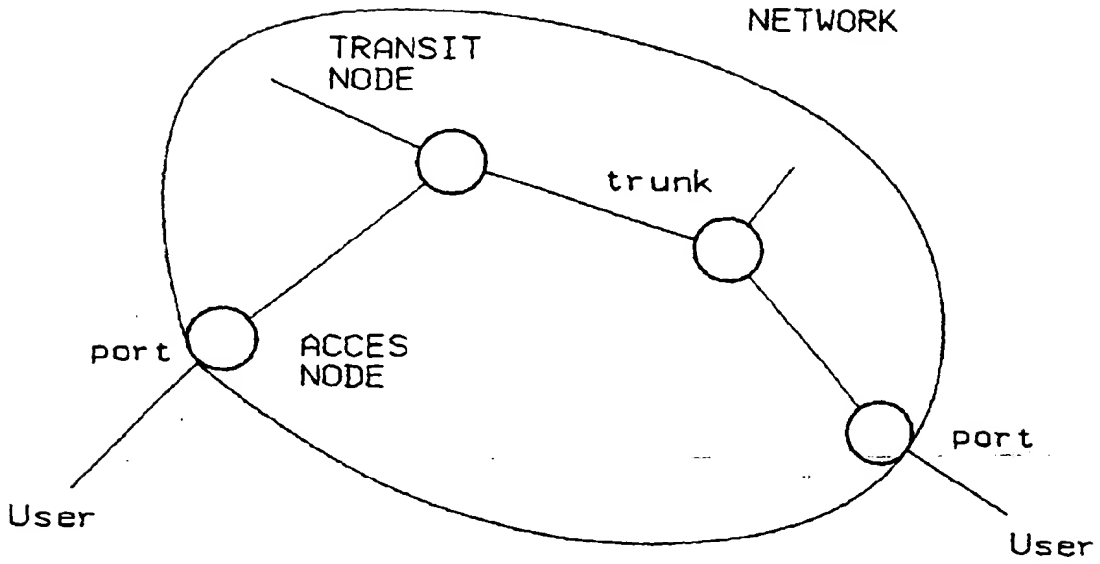


FIGURE 2

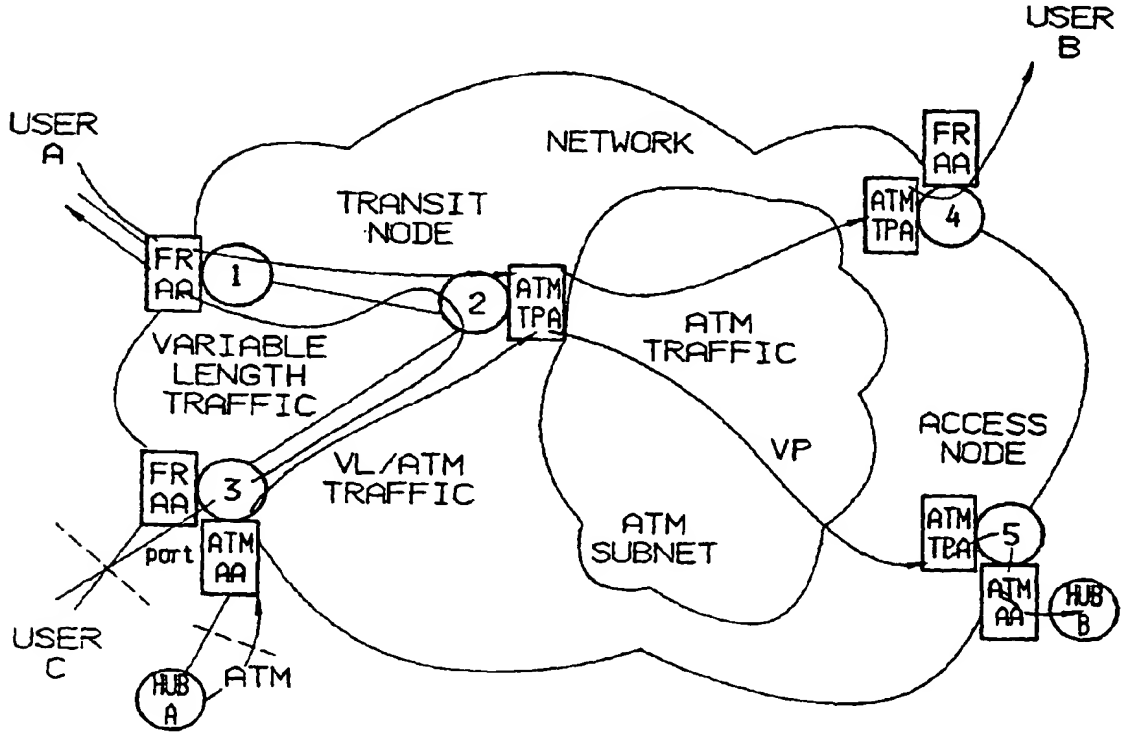


FIGURE 3

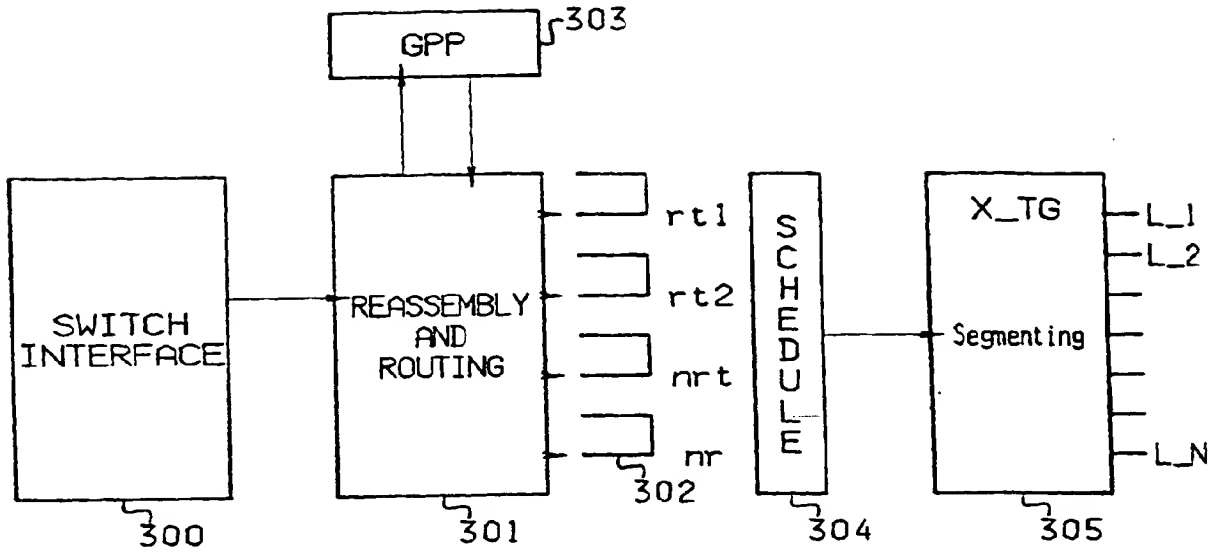


FIGURE 4

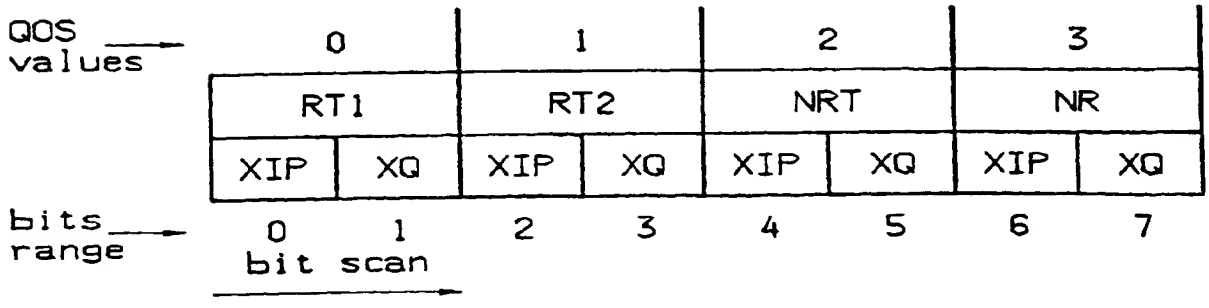


FIGURE 5

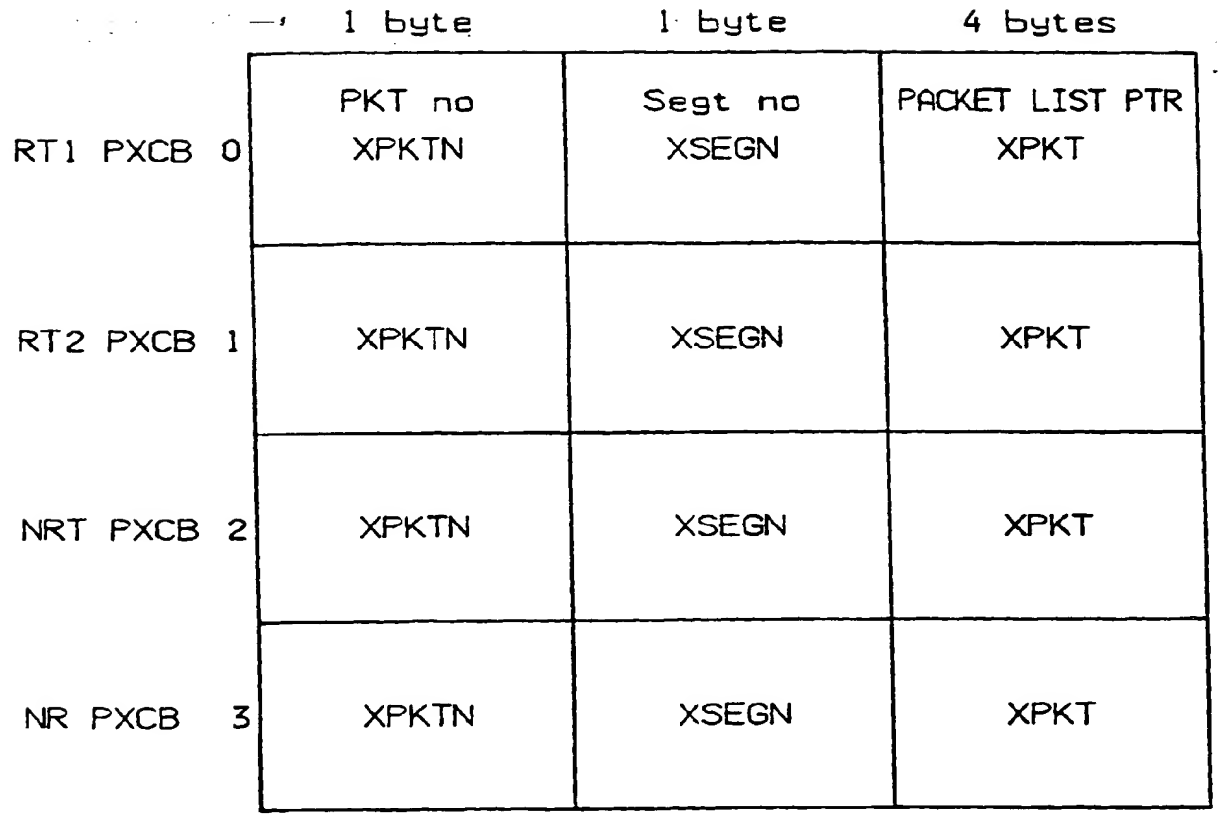


FIGURE 6

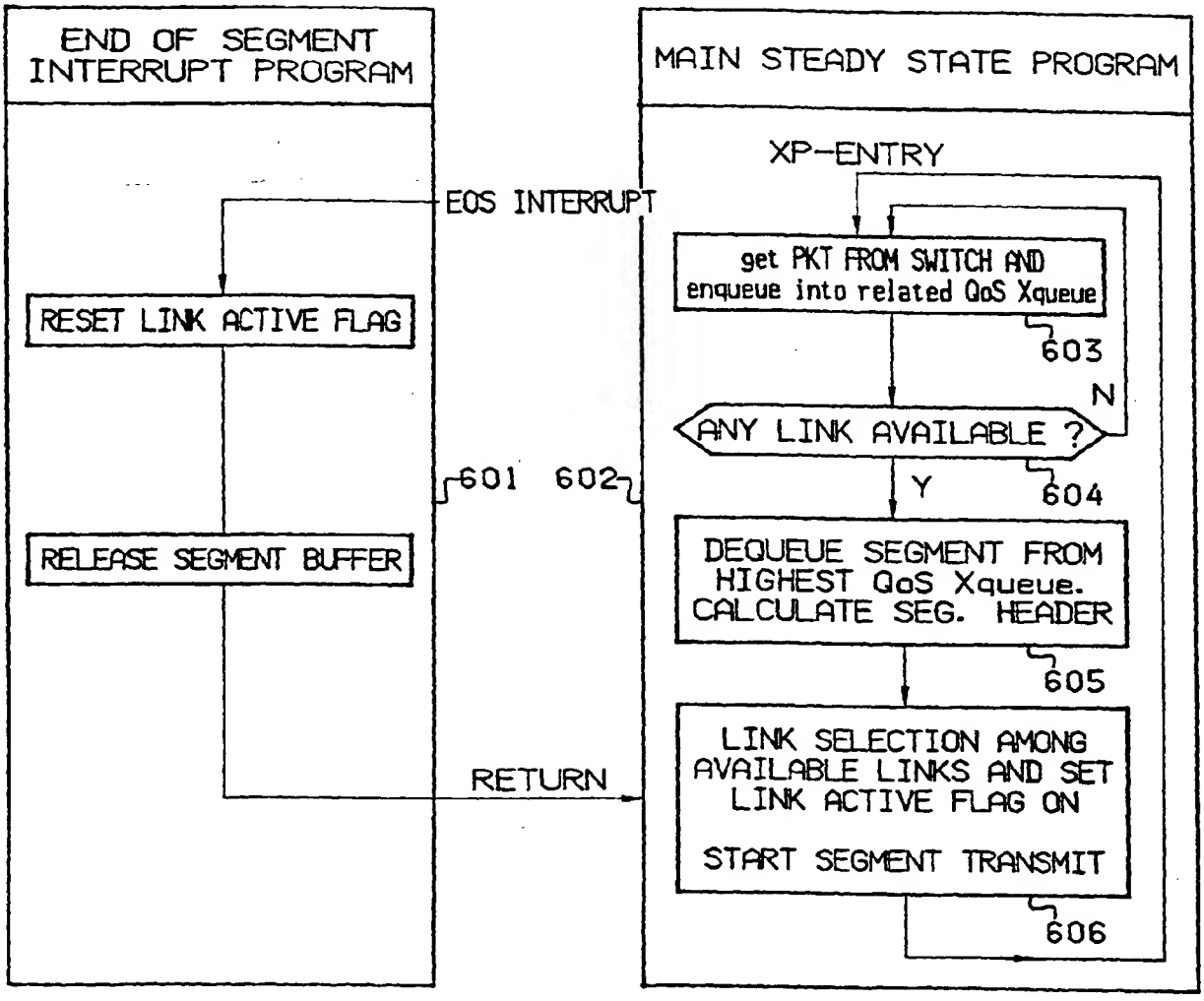


FIGURE 7

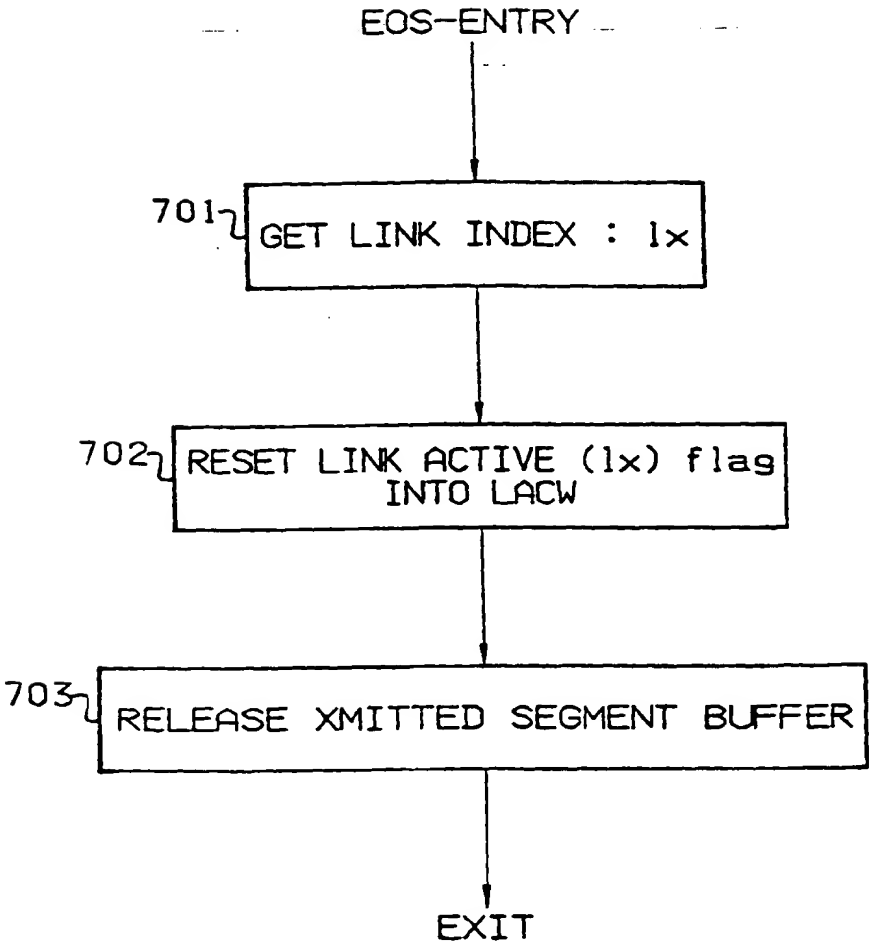


FIGURE 8

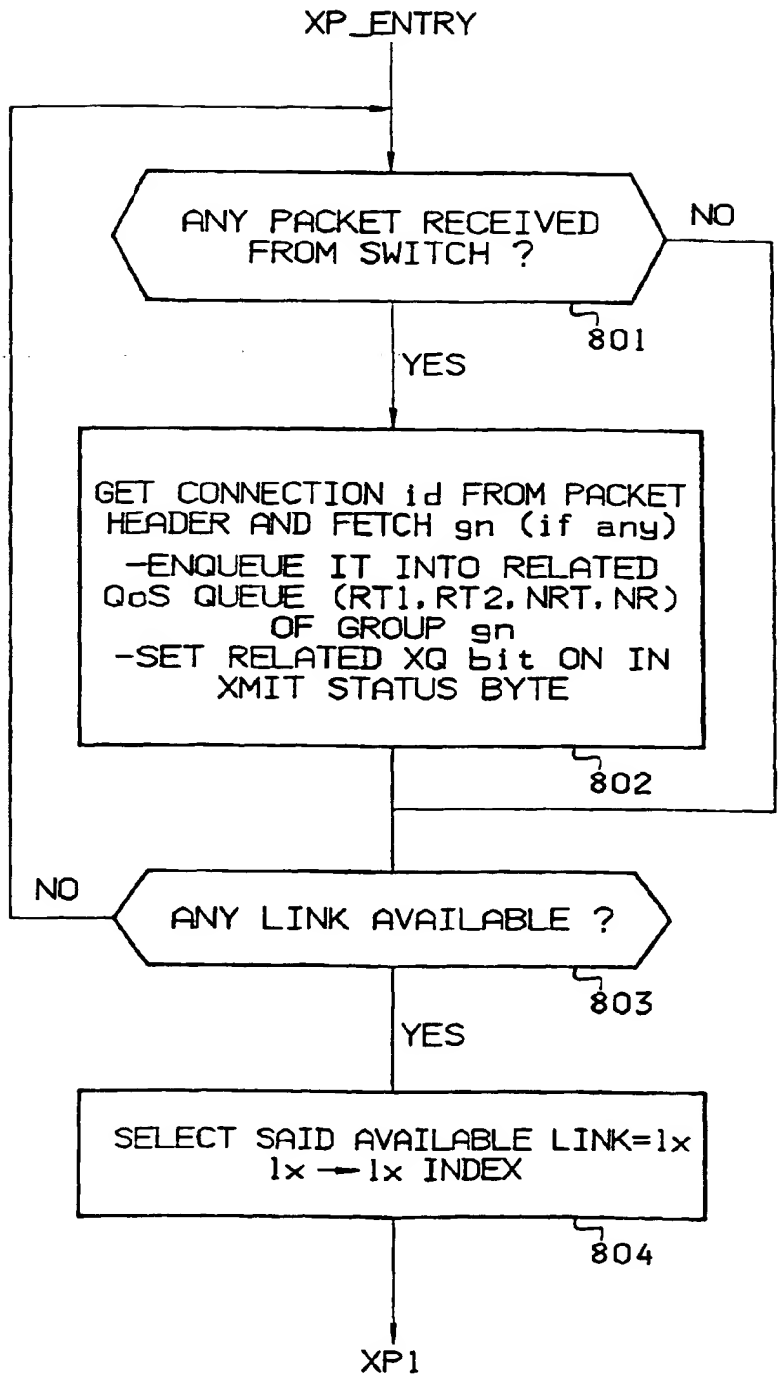


FIGURE 9

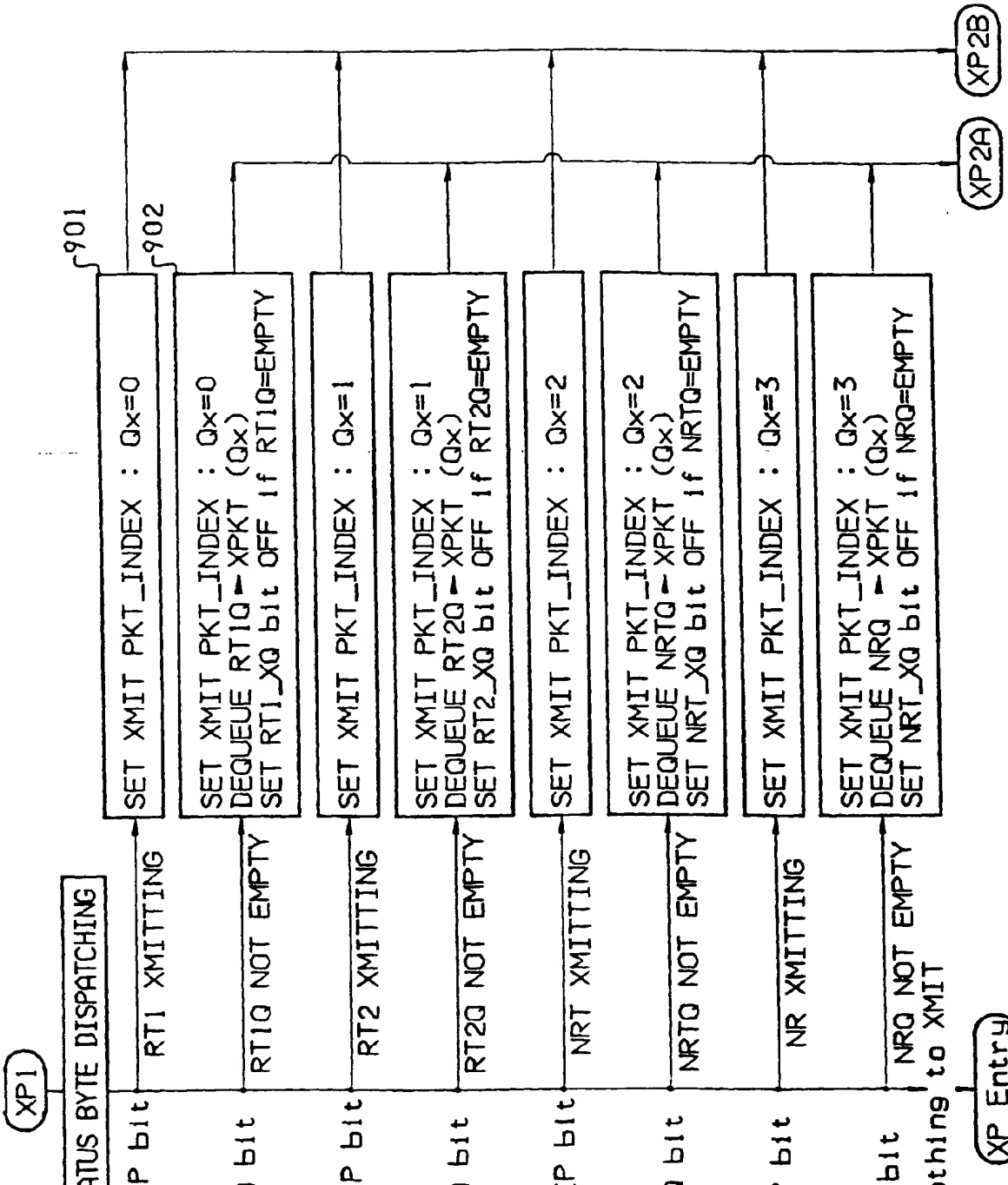


FIGURE 12

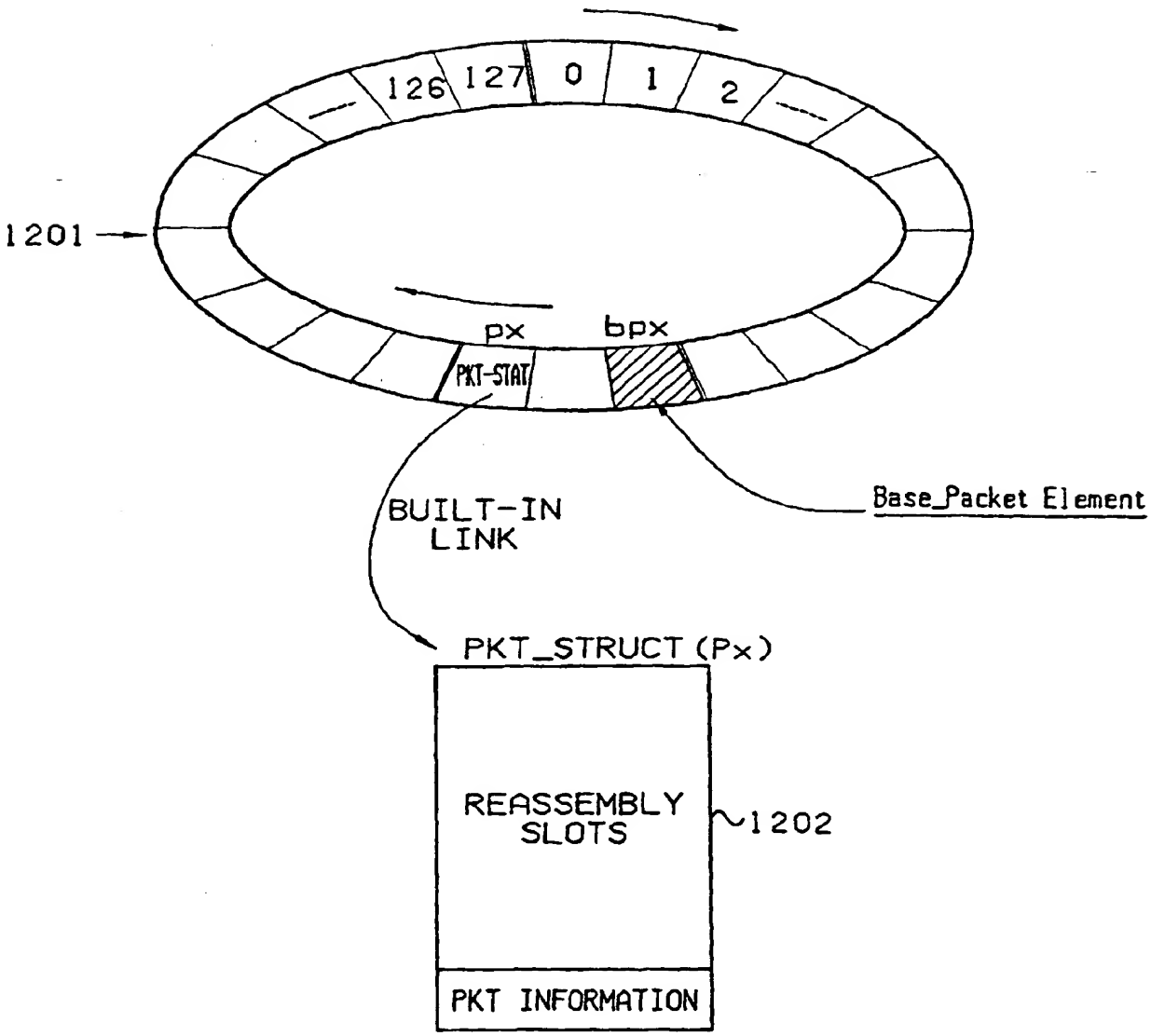


FIGURE 13

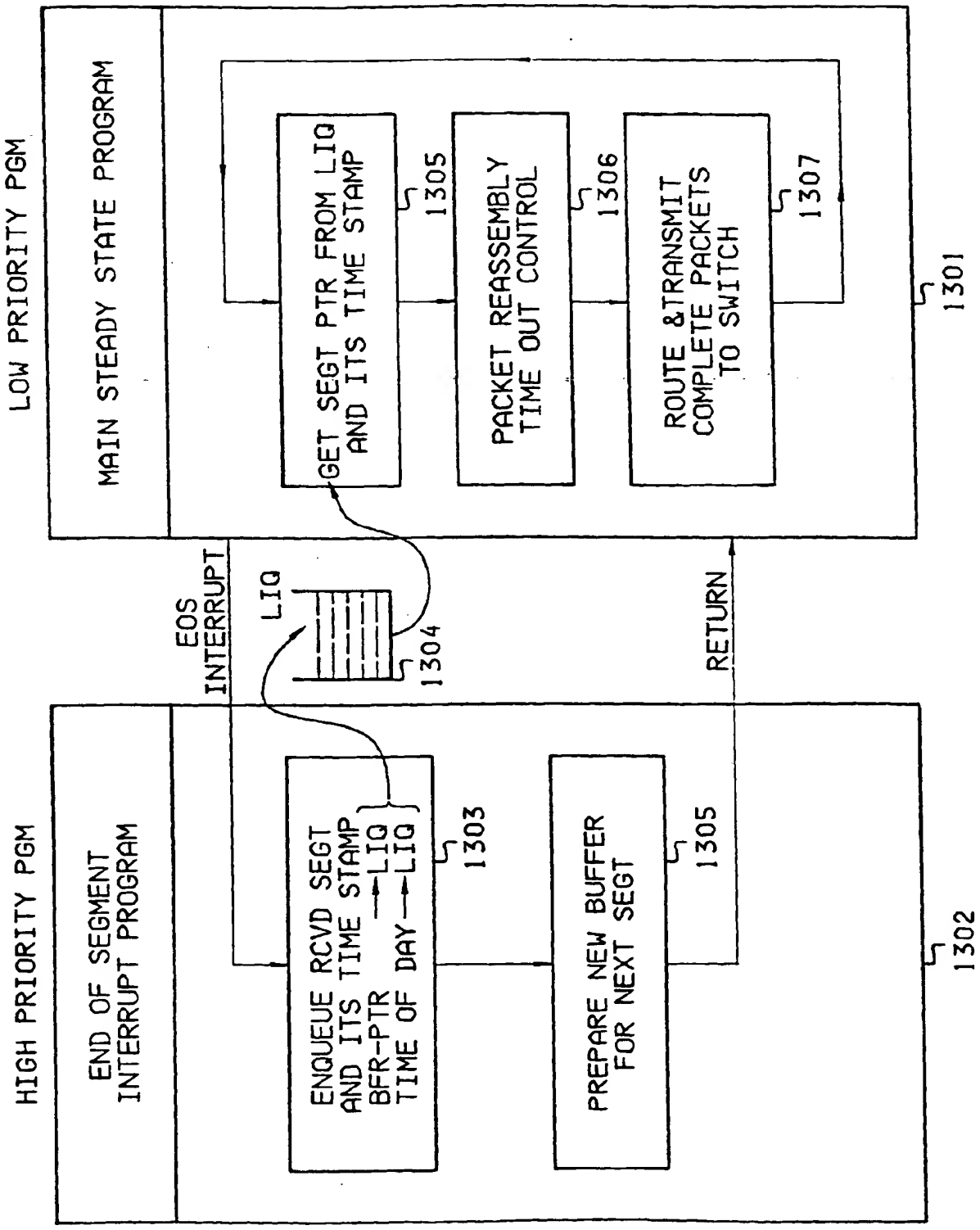

LOW PRIORITY PGM
MAIN STEADY STATE PROGRAM
GET SEGT PTR FROM LIQ AND ITS TIME STAMP
1305
PACKET REASSEMBLY TIME OUT CONTROL
1306
ROUTE &TRANSMIT COMPLETE PACKETS TO SWITCH
1307
1301
EOS INTERRUPT
LIQ
1304
RETURN
HIGH PRIORITY PGM
END OF SEGMENT INTERRUPT PROGRAM
ENQUEUE RCVD SEGT AND ITS TIME STAMP
BFR-PTR →LIQ
TIME OF DAY →LIQ
1303
PREPARE NEW BUFFER FOR NEXT SEGT
1305
1302

FIGURE 14

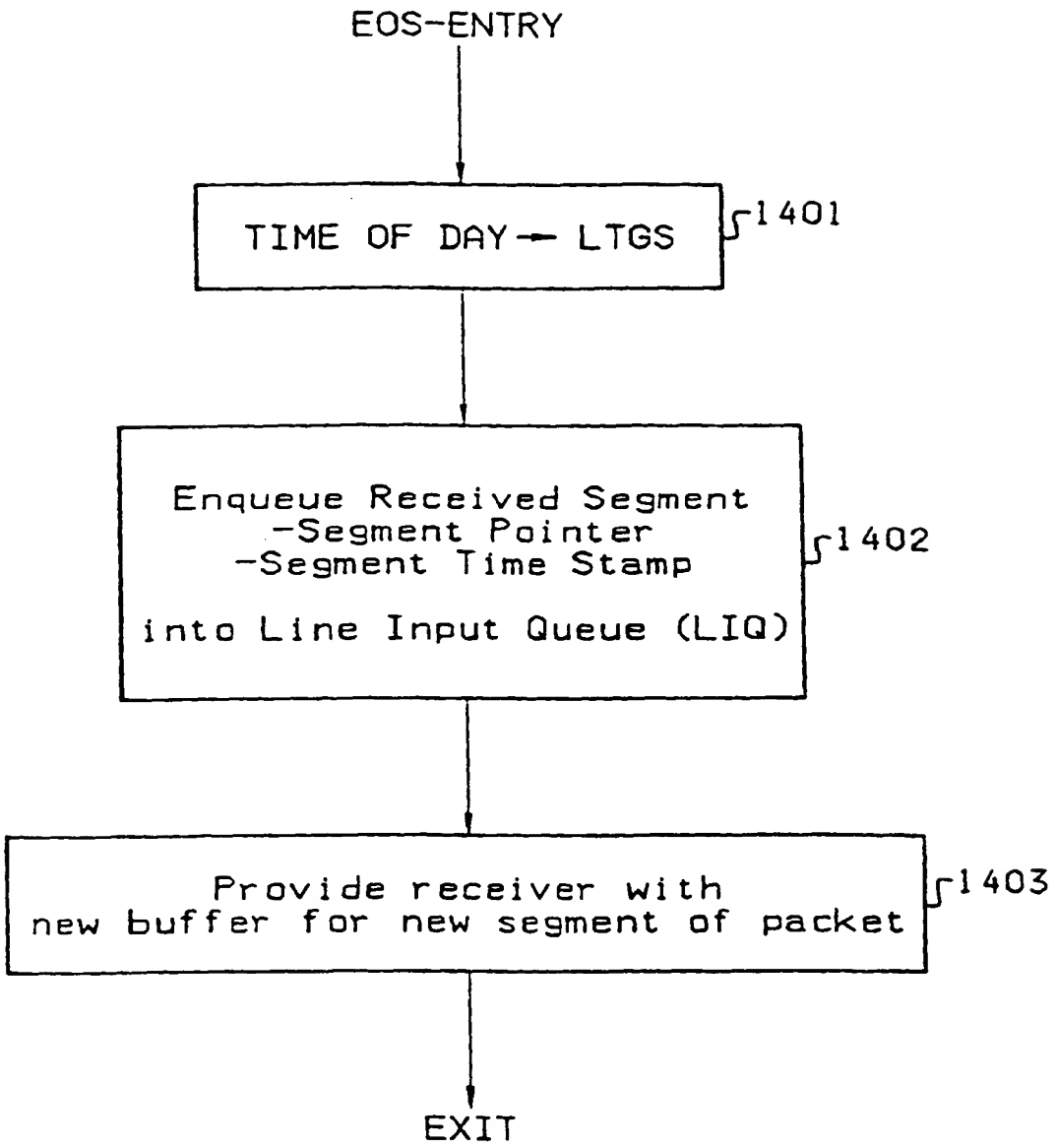


FIGURE 15

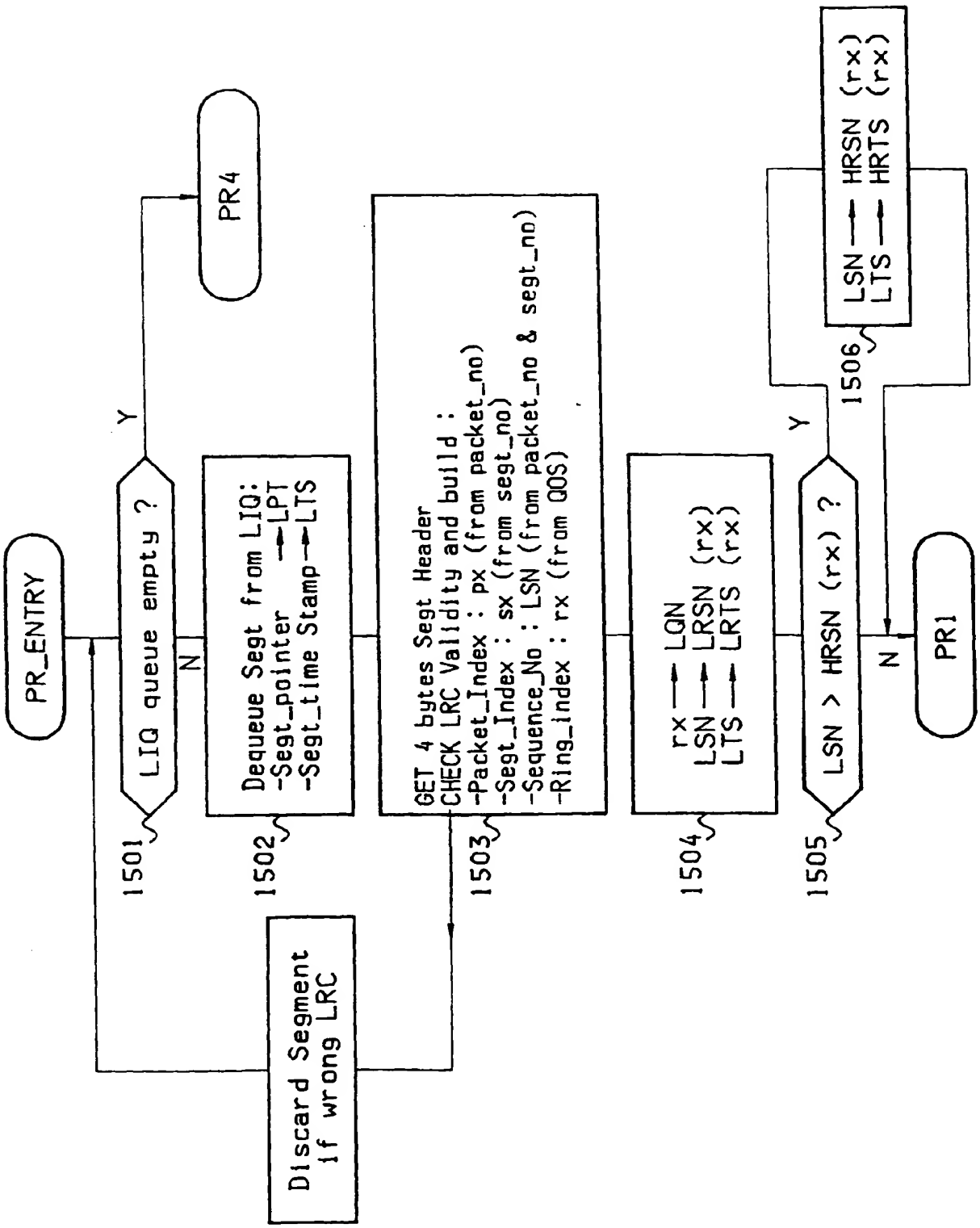


FIGURE 16

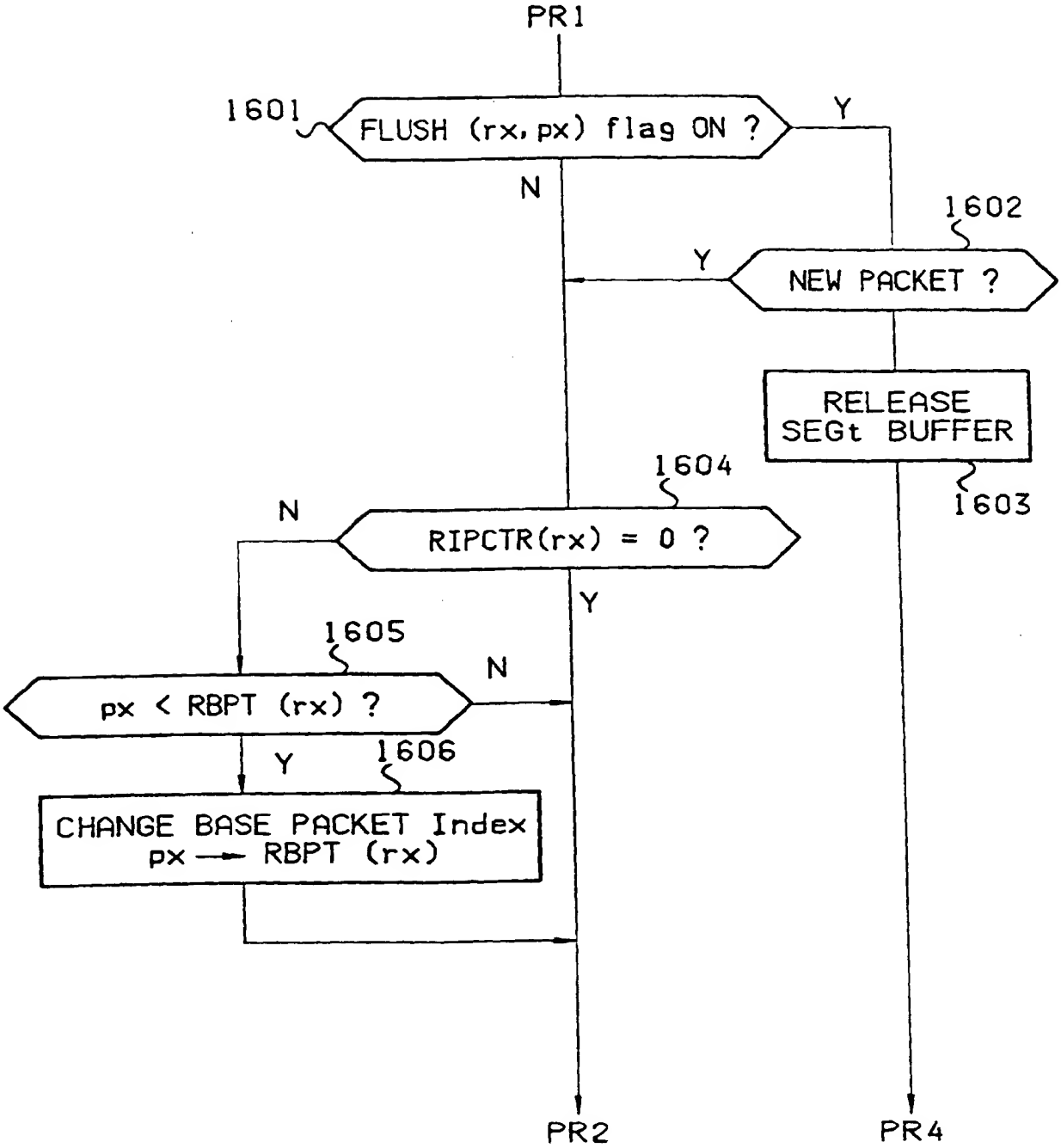


FIGURE 17

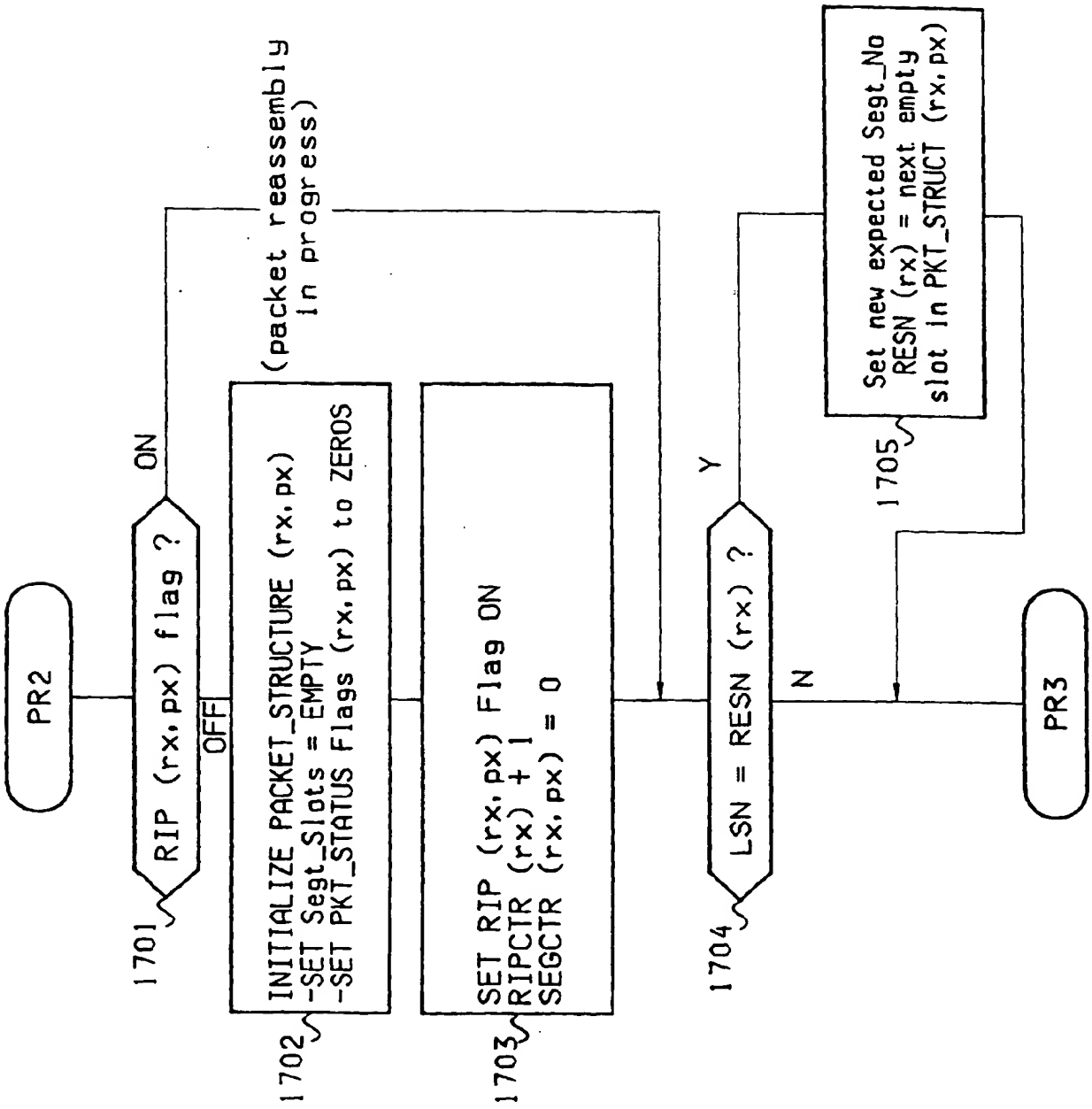


FIGURE 18

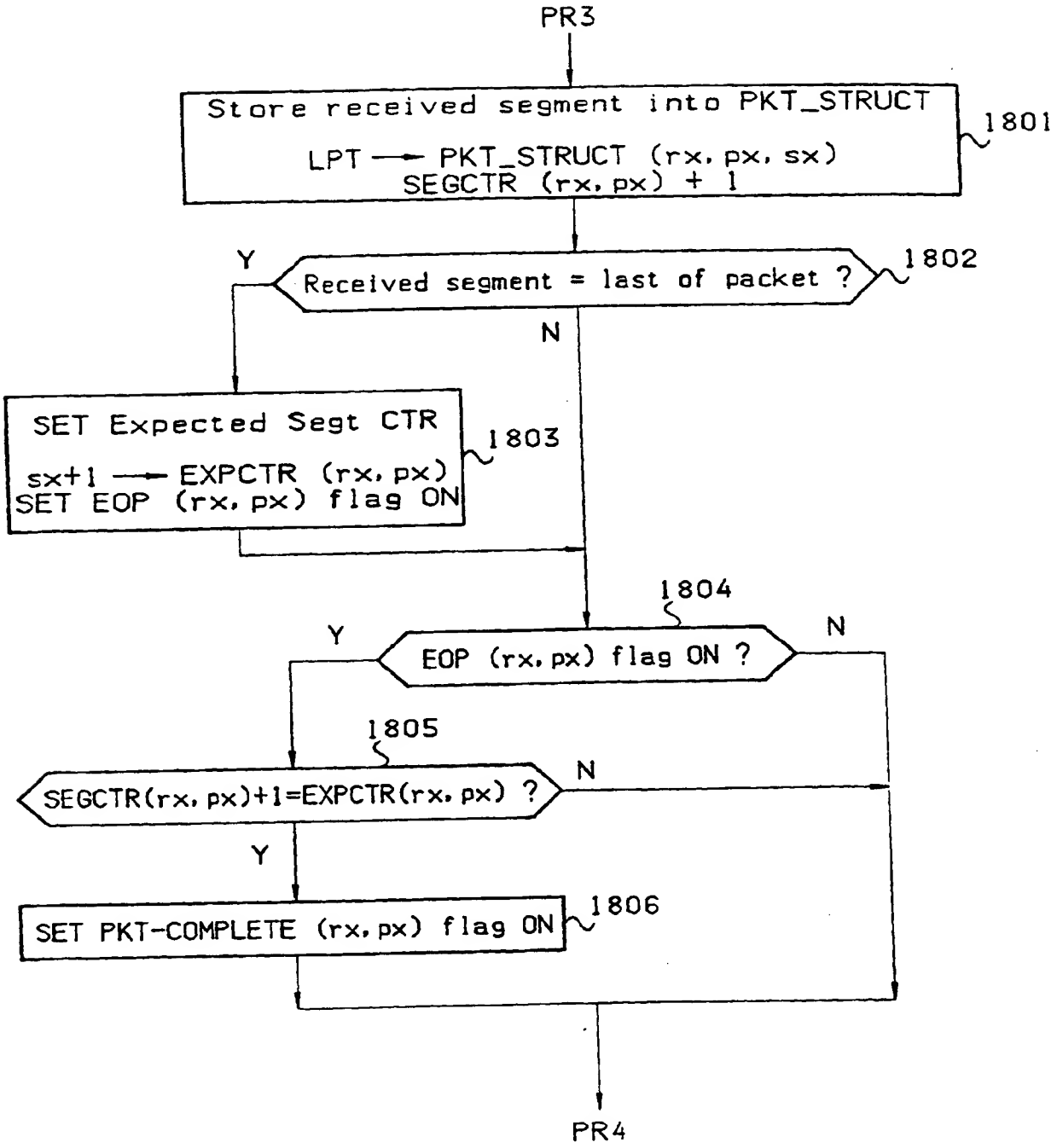


FIGURE 19

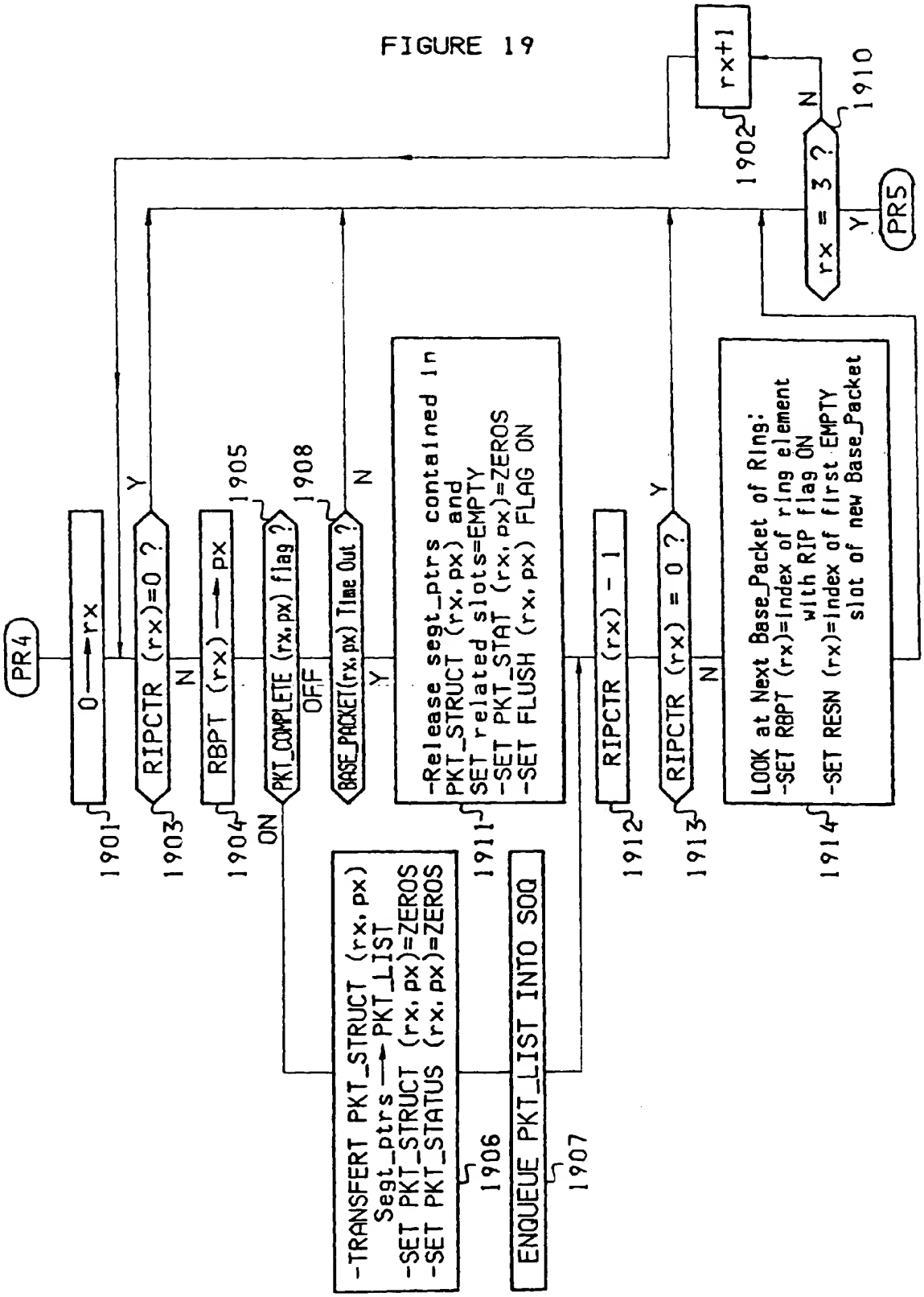


FIGURE 20

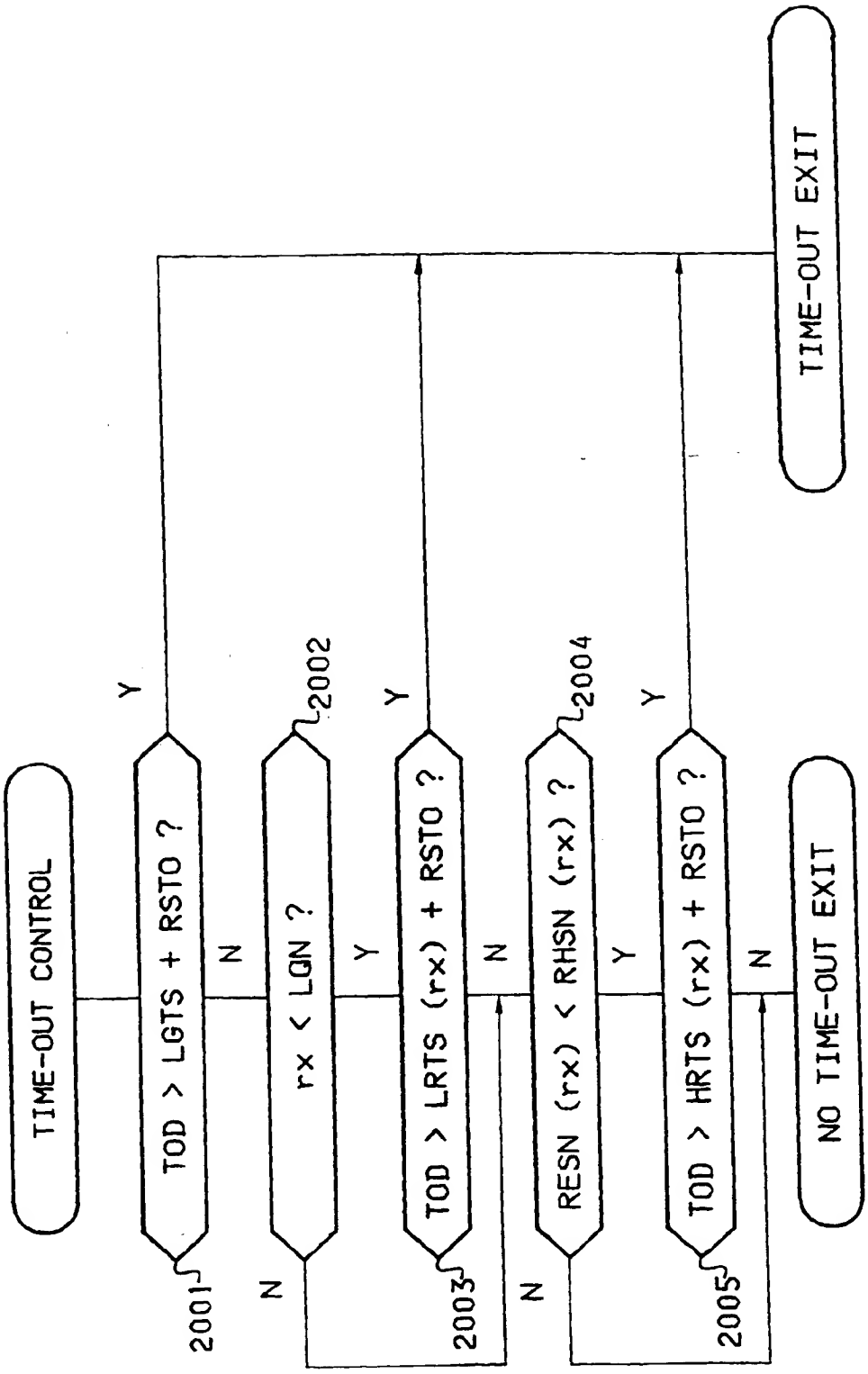
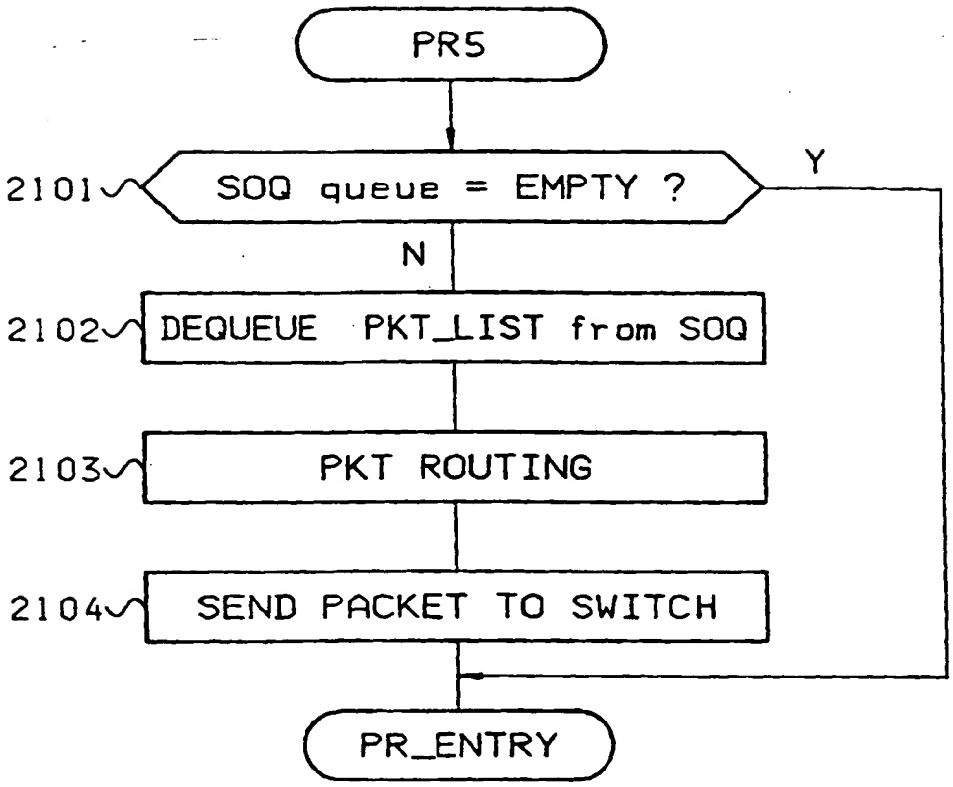


FIGURE 21



(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 855 820 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.07.1998 Bulletin 1998/31

(51) Int Cl.[6]: **H04L 12/56, H04Q 11/04**

(21) Application number: **97480085.6**

(22) Date of filing: **28.11.1997**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

[30] Priority: **13.12.1996 EP 96480111**

(71) Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION**
Armonk, NY 10504 (US)

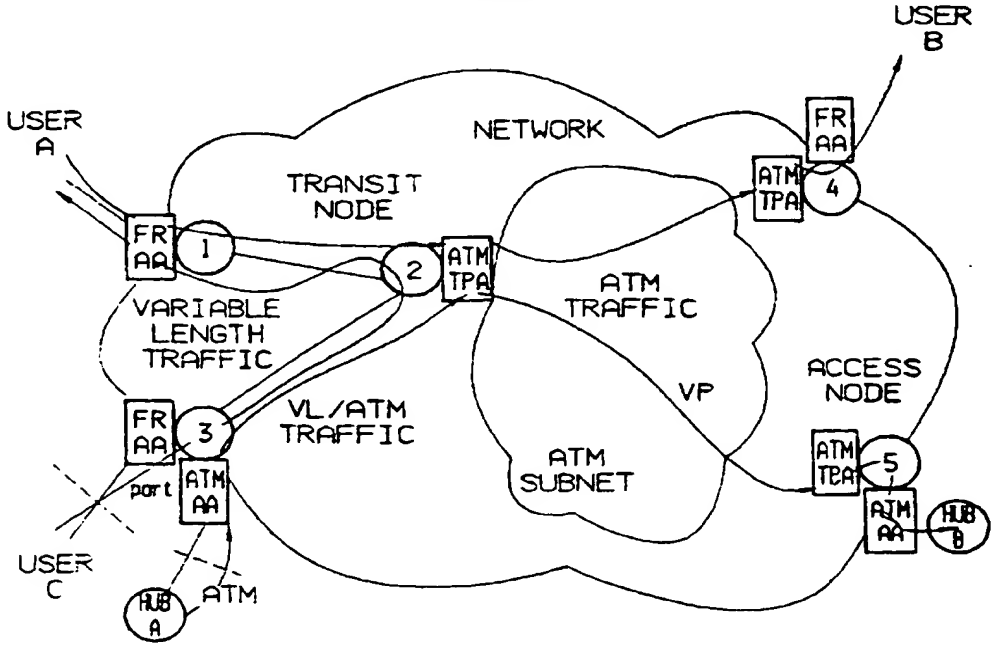
(72) Inventors:
- **Galand, Claude**
  06480 La Colle/Loup (FR)
- **Spagnol, Victor**
  06800 Cagnes sur Mer (FR)
- **Lebizay, Gérald**
  06140 Vence (FR)

(74) Representative: **Therias, Philippe**
Compagnie IBM FRANCE,
Département de Propriété Intellectuelle
06610 La Gaude (FR)

(54) A method and system for optimizing data transmission line bandwidth occupation in a multipriority data traffic environment

(57) This invention deals with data communication networks and more particularly with a method and system for optimizing data link occupation in a multipriority data traffic environment by using data multiplexing techniques over fixed or variable length data packets being asynchronously transmitted. Said packets are split into segments including both a segment number and a packet number. Then the segments are dispatched, on priority basis, over available links or virtual channels based on a so-called global link availability control word indications, which control word is dynamically adjusted according to specific predefined conditions.

FIGURE 2



EP 0 855 820 A2 CORRIGENDUM Issued on 04.11.1998 (bibliography updates included)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 855 820 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
16.04.2003 Bulletin 2003/16

(43) Date of publication A2:
29.07.1998 Bulletin 1998/31

(21) Application number: **97480085.6**

(22) Date of filing: **28.11.1997**

(51) Int Cl.[7]: **H04L 12/56**, H04Q 11/04

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **13.12.1996 EP 96480111**

(71) Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION**
Armonk, NY 10504 (US)

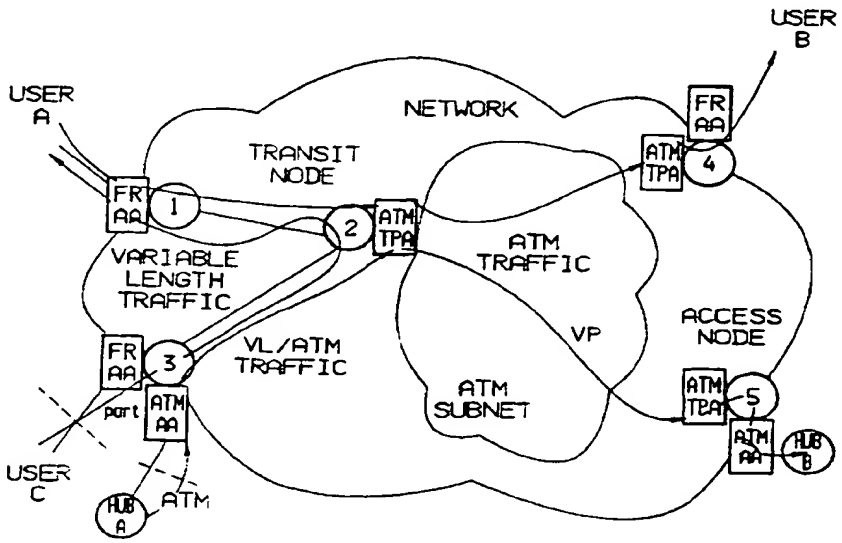
(72) Inventors:
- **Galand, Claude**
  06480 La Colle/Loup (FR)
- **Spagnol, Victor**
  06800 Cagnes sur Mer (FR)
- **Lebizay, Gérald**
  06140 Vence (FR)

(74) Representative: **Therias, Philippe**
Compagnie IBM FRANCE,
Département de Propriété Intellectuelle
06610 La Gaude (FR)

(54) A method and system for optimizing data transmission line bandwidth occupation in a multipriority data traffic environment

(57) This invention deals with data communication networks and more particularly with a method and system for optimizing data link occupation in a multipriority data traffic environment by using data multiplexing techniques over fixed or variable length data packets being asynchronously transmitted. Said packets are split into segments including both a segment number and a packet number. Then the segments are dispatched, on priority basis, over available links or virtual channels based on a so-called global link availability control word indications, which control word is dynamically adjusted according to specific predefined conditions.

FIGURE 2



EP 0 855 820 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 48 0085

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	WO 96 08120 A (STRATACOM INC) 14 March 1996 (1996-03-14) * abstract * * page 10, paragraph 2 - page 11, paragraph 1; figure 2 * * page 15, paragraph 2 - page 19, paragraph 1; figure 8; tables 1-5 *	1,4	H04L12/56 H04Q11/04
A	WO 96 17489 A (NORTHERN TELECOM LTD) 6 June 1996 (1996-06-06) * abstract * * page 7, line 16 - page 11, line 6; figures 2,4 *	1,4	
A	FREDETTE P H: "THE PAST, PRESENT, AND FUTURE OF INVERSE MULTIPLEXING" IEEE COMMUNICATIONS MAGAZINE, IEEE SERVICE CENTER. PISCATAWAY, N.J, US, vol. 32, no. 4, 1 April 1994 (1994-04-01), pages 42-46, XP000451028 ISSN: 0163-6804 * the whole document *		

TECHNICAL FIELDS SEARCHED (Int.Cl.6)

H04Q
H04L

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	18 February 2003	von der Straten, G

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 97 48 0085

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

18-02-2003

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9608120 A	14-03-1996	US 5617417 A	01-04-1997
		AU 706853 B2	24-06-1999
		AU 3321495 A	27-03-1996
		CA 2199383 A1	14-03-1996
		EP 0780046 A1	25-06-1997
		WO 9608120 A1	14-03-1996
		US 5970067 A	19-10-1999
WO 9617489 A	06-06-1996	US 5608733 A	04-03-1997
		AT 173373 T	15-11-1998
		CA 2204171 A1	06-06-1996
		WO 9617489 A1	06-06-1996
		DE 69506003 D1	17-12-1998
		DE 69506003 T2	15-04-1999
		EP 0795259 A1	17-09-1997
		JP 10500271 T	06-01-1998
		JP 3087182 B2	11-09-2000

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82